

INTERNATIONAL SEARCH REPORT

Intern Application No

PCT/EP 00/08543

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N1/41

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 956 467 A (HONSINGER CHRIS W ET AL) 21 September 1999 (1999-09-21) abstract; claims	1
A	WO 96 15620 A (UNIV HONG KONG CHINESE ;WU XIAOLIN (CA); MEMON NASIR (US)) 23 May 1996 (1996-05-23) abstract	1
A	DE 198 04 928 A (EASTMAN KODAK CO) 3 September 1998 (1998-09-03) the whole document	1
	--- -/--	

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- * & * document member of the same patent family

Date of the actual completion of the international search

14 December 2000

Date of mailing of the international search report

22/12/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040. Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Isa, S

INTERNATIONAL SEARCH REPORT

Interr. Appl. Application No
PCT/EP 00/08543

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PATENT ABSTRACTS OF JAPAN vol. 1997, no. 11, 28 November 1997 (1997-11-28) & JP 09 200532 A (RICOH CO LTD), 31 July 1997 (1997-07-31) abstract ---	1
A	PATENT ABSTRACTS OF JAPAN vol. 1998, no. 09, 31 July 1998 (1998-07-31) & JP 10 098718 A (RICOH CO LTD), 14 April 1998 (1998-04-14) abstract ---	1
A	PATENT ABSTRACTS OF JAPAN vol. 011, no. 382 (E-564), 12 December 1987 (1987-12-12) & JP 62 145969 A (FUJITSU LTD), 30 June 1987 (1987-06-30) abstract -----	1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/EP 00/08543

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5956467 A	21-09-1999	NONE	
WO 9615620 A	23-05-1996	US 5903676 A	11-05-1999
DE 19804928 A	03-09-1998	JP 10257330 A	25-09-1998
JP 09200532 A	31-07-1997	NONE	
JP 10098718 A	14-04-1998	NONE	
JP 62145969 A	30-06-1987	NONE	

THIS PAGE BLANK (USPTO)

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro



(43) Internationales Veröffentlichungsdatum
22. März 2001 (22.03.2001)

PCT

(10) Internationale Veröffentlichungsnummer
WO 01/20897 A1

(51) Internationale Patentklassifikation⁷: H04N 1/41

(21) Internationales Aktenzeichen: PCT/EP00/08543

(22) Internationales Anmeldedatum:
1. September 2000 (01.09.2000)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
199 44 213.4 15. September 1999 (15.09.1999) DE

(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme
von US): INSTITUT FÜR TELEMATIK E.V. [DE/DE];
Bahnhofstrasse 30-32, 54292 Trier (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder (nur für US): MEINEL, Christoph
[DE/DE]; Am Hohlweg 16, 54317 Gusterath (DE).
KHLOUDOV, Serguei [DE/DE]; Kleeburger Weg 2,
54296 Trier (DE).

(74) Anwalt: FRIZ, Oliver; Dreiss, Fuhlendorf, Steimle &
Becker, Postfach 10 37 62, 70032 Stuttgart (DE).

(81) Bestimmungsstaaten (national): CA, CN, KR, RU, US.

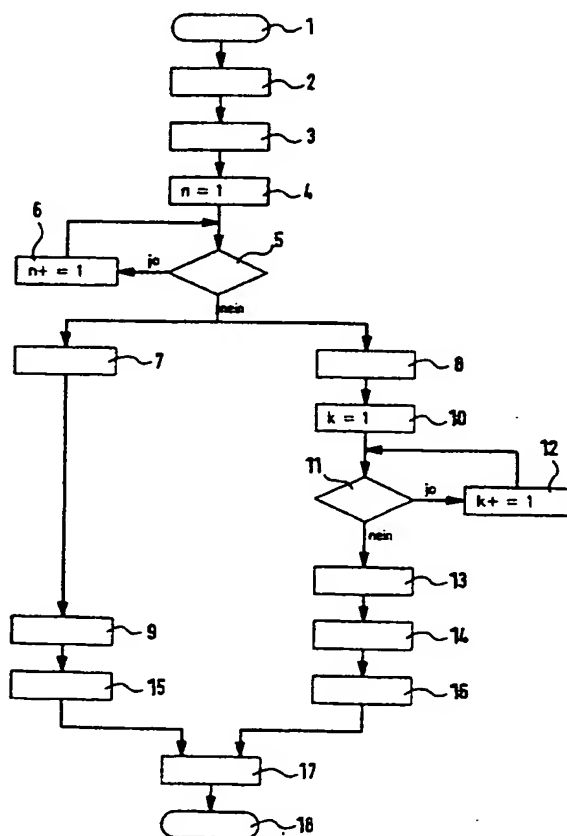
(84) Bestimmungsstaaten (regional): europäisches Patent (AT,
BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC,
NL, PT, SE).

Veröffentlicht:
— Mit internationalem Recherchenbericht.

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR COMPRESSING A DIGITAL IMAGE WITH SEVERAL BIT-PLANES

(54) Bezeichnung: VERFAHREN ZUM KOMPRIMIEREN EINES DIGITALEN BILDES MIT MEHREREN BIT-EBENEN



(57) Abstract: The invention relates to a method for compressing a digital image (DB), whose image information is encoded on several bit planes (BE). The invention provides an adaptive compression method which allows the most efficient compression possible for digital images with any configuration. Said method comprises the following steps: Analysis of the bit planes (BE) of the digital image (DB); Sub-division of the bit planes (BE) of the digital image (DB) into several bit-plane areas, each containing at least one bit plane (BE); Division of the digital image (DB) into several bit-plane area images (BEBB), each containing one of the bit-plane areas (block 7; block 8); Selection of a specific compression method for each bit-plane area image (BEBB); and Compression of the individual bit-plane area images (BEBB) using the respective compression method which has been selected (block 15; block 16).

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zum Komprimieren eines digitalen Bildes (DB), dessen Bildinformationen in mehreren Bitebenen (BE) codiert sind. Es wird ein adaptives Komprimierungsverfahren vorgeschlagen, das für beliebig ausgestaltete digitale Bilder eine möglichst effiziente Komprimierung ermöglicht. Es umfasst die nachfolgenden Schritte: Analysieren der Bitebenen (BE) des digitalen Bildes (DB); Unterteilen der Bitebenen (BE) des digitalen Bildes (DB) in mehrere Bitebenenbereiche, die jeweils mindestens eine Bitebene (BE) aufweisen; Aufteilen des digitalen Bildes (DB) in mehrere Bitebenenbereichsbilder (BEBB), die jeweils einen der Bitebenenbereiche umfassen (Block 7; Block 8); Auswählen eines bestimmten Komprimierungsverfahrens für jedes Bitebenenbereichsbild (BEBB); und Komprimieren der einzelnen Bitebenenbereichsbilder (BEBB) mit dem jeweils ausgewählten Komprimierungsverfahren (Block 15; Block 16).

WO 01/20897 A1

WO 01/20897 A1



Zur Erklärung der Zweibuchstaben-Codes, und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

Verfahren zum Komprimieren eines digitalen Bildes mit mehreren Bit-Ebenen

Beschreibung

Die vorliegende Erfindung betrifft ein Verfahren zum Komprimieren eines digitalen Bildes, dessen Bildinformationen in mehreren Bitebenen codiert sind. Die Erfindung betrifft des Weiteren ein Verfahren zum Übertragen eines digitalen Bildes, dessen Bildinformationen in mehreren Bitebenen codiert sind, von dem Endgerät eines Senders zu dem Endgerät eines Empfängers.

Aus dem Stand der Technik sind ein Vielzahl unterschiedlicher Verfahren bekannt, um digitale Bilder, insbesondere Fotos und Zeichnungen, in einer Bilddatei zu speichern, die von einem Computer gelesen werden kann. Alle diese bekannten Verfahren nutzen Komprimierungsverfahren aus, um die Bilddatei so klein wie möglich zu machen, damit sie möglichst wenig Speicherplatz in dem Hauptspeicher und auf Speichermedien des Computers benötigt. Außerdem lassen sich digitale Bilder umso schneller von dem Endgerät eines Senders zu dem Endgerät eines Empfängers übertragen, desto kleiner die Bilddatei ist. Als Endgeräte werden üblicherweise Computer eingesetzt, es ist aber auch denkbar, mit der notwendigen Hard- und Software ausgestattete Telefone o. ä. zu verwenden. Die Komprimierung digitaler Bilder ist insbesondere in der modernen Medizintechnik von Bedeutung, wo die Abbildungen moderner Computerdiagnostikgeräte (z. B. Kernspin-Tomograph, Röntgengerät, Computer-Tomograph, etc.) zunehmend in digitaler Form zum Abspeichern oder Übertragen an andere Computer vorliegen.

Einige aus dem Stand der Technik bekannten Verfahren zum Komprimieren eines digitalen Bildes sind bspw. das Windows-Bit-Map (BMP)-Format, das insbesondere von kleineren, unter

Windows lauffähigen Graphikprogrammen verwendet wird. Das PCX-Format und das Windows-Metafile (WMF)-Format wird ebenfalls von Windows-Anwendungen, insbesondere für Cliparts und andere Abbildungen verwendet. Das Tagged-Image-File-Format (TIF)-Verfahren wird von einer Vielzahl von Graphikprogrammen zum Abspeichern digitaler Bilder verwendet. Das Graphics-Interchange-Format (GIF)-Verfahren und das Joint-Photographic-Experts-Group (JPEG)-Verfahren wird insbesondere im Internet zum Übertragen von digitalen Bildern verwendet. Diese bekannten Komprimierungsverfahren stehen in der Regel in modernen Browsern, die zum Übertragen von Dateien zwischen Computern eingesetzt werden, zur Verfügung und können bei Bedarf eingesetzt werden.

Das BMP-, WMF-, PIF-, GIF-Format wird für die Bildbeschreibung angewendet. Für die Kompression wird das Huffman-, Shannon-Fano-, Arithmetik-, LZW-, LZ77-, LZ78-, JPEG-, Fractal-, Scalar- oder Vektor-Quantisierungsverfahren angewandt.

Einige weniger verbreitete Formate zur Bilddarstellung eines digitalen Bildes umfassen das Computer-Graphics-Metafile (CGM)-Verfahren, das Encapsulated-Postscript (EPSI/EPSP)-Verfahren, das Group-4-Type-I (G4 CCITT)-Verfahren, das Portable-Bitmap (PBM)-Verfahren, das Portable-Greymap (PGM)-Verfahren, das Portable-Network-Graphics (PNG)-Verfahren, das Portable-Pixmap (PPM)-Verfahren, das Silicon-Graphics (RGB)-Verfahren, das Sun-Raster (SUN)-Verfahren und das Group-4-Type-II (TG4 CCITT)-Verfahren. Ausführliche Informationen zum Thema Komprimieren von digitalen Bildern können dem Internet auf der Seite <http://www.cs.cmu.edu/~guyb/real-world/compress/index.html> (Stand 13.09.1999, zuletzt geändert am 21.07.1999) entnommen werden.

Jedes dieser bekannten Verfahren zum Komprimieren eines digitalen Bildes erzielt bei bestimmten digitalen Bildern einen besonders hohen Kompressionsrate, während es für andere Bilder weniger gut geeignet ist. Das GIF-Verfahren wird

vorzugsweise für Zeichnungen oder künstlerische Abbildungen verwendet. Bei dem GIF-Verfahren werden die digitalen Bilder über einen Wechsel der Farb- oder Graustufeninformationen und nicht über die tatsächlichen Farben oder Graustufen komprimiert. D.h. die komprimierte Bilddatei ist umso kleiner, je weniger Farb- und Graustufenwechsel das digitale Bild enthält, d.h. desto größere Flächen derselben Farbe oder Graustufe es enthält. Das GIF-Verfahren arbeitet mit einer 8-bit Auflösung (Farb- oder Graustufentiefe) pro Pixel. Aufgrund dieser relativ geringen Auflösung kann es, insbesondere bei fotografischen Abbildungen, bei der komprimierten Bilddatei dazu kommen, dass bestimmte Farben oder Graustufen verschwommen sind oder aus mehreren Farben oder Graustufen der zur Verfügung stehenden Palette aufgebaut sind, was für einen Betrachter des komprimierten Bildes in einer relativ groben Körnung des Bildes zu erkennen ist. Da das GIF-Verfahren die Informationen eines digitalen Bildes über die Wechsel der Farben oder Graustufen speichert, kann es beim Komprimieren einer fotografischen Abbildung mit hochauflösenden Wechseln der Farben oder Graustufen oder mit vielen Farben oder Graustufen zu einem starken Anstieg der Größe der komprimierten Bilddatei kommen. Nähere Informationen zu dem GIF-Verfahren können im Internet auf der Seite <http://www.rit.edu/~mpb9954/mmwww/GIFComp.htm> (Stand 01.09.1999; zuletzt geändert 22.09.1997) und der Seite <http://www.cs.cmu.edu/~guyb/real-world/compress/index.html> (Stand 13.09.1999, zuletzt geändert am 21.07.1999) entnommen werden.

Das JPEG-Verfahren wird vorzugsweise für fotografische oder andere hochauflösende Abbildungen verwendet. Im Gegensatz zu dem GIF-Verfahren führt das JPEG-Verfahren bei einer hohen Feindetaillierung zu einer relativ hohen Kompressionsrate. Allerdings arbeitet das JPEG-Verfahren bei digitalen Bildern, die große Flächen derselben Farbe oder Graustufe enthalten nicht so effizient, da es zusätzliche Informationen innerhalb und um diese Flächen herum hinzufügt, was die Abbildung

unsauber aussehen lässt. Nähere Informationen über das JPEG-Verfahren können dem Internet auf der Seite <http://www.rit.edu/~mpb9954/mmwww/JPGComp.htm> (Stand 01.09.1999, zuletzt geändert am 22.09.1997) und der Seite <http://www.cs.cmu.edu/~guyb/real-world/compress/index.html> (Stand 13.09.1999, zuletzt geändert am 21.07.1999) entnommen werden.

Schon der Vergleich dieser beiden, aus den bekannten Komprimierungsverfahren beispielhaft herausgegriffenen Komprimierungsverfahren macht deutlich, dass bestimmte Komprimierungsverfahren bei bestimmten digitalen Bildern eine bessere Kompressionsrate erzielen können als bei anderen digitalen Bildern. Da digitale Bilder in der Regel beliebig ausgestaltete Bildbereiche (z. B. große Flächen derselben Farbe oder Graustufe einerseits und hochauflösende Wechsel und viele Farben oder Graustufen andererseits) aufweisen, stellt die Auswahl eines einzigen Komprimierungsverfahrens für ein digitales Bild in der Regel nur einen Kompromiss dar.

Aus diesem Grund werden im Stand der Technik sogenannte adaptive Komprimierungsverfahren eingesetzt, die die Bildbereiche eines digitalen Bildes auf seine Ausgestaltung hin analysieren und in Abhängigkeit von den Analyseergebnissen für verschiedene Bereiche des Bildes unterschiedliche Komprimierungsverfahren anwenden.

Aus dem Stand der Technik sind drei verschiedene adaptive Komprimierungsverfahren bekannt:

Die erste Gruppe adaptiver Komprimierungsverfahren beinhaltet lokale adaptive Algorithmen. Das digitale Bild wird zunächst in nicht überlappende Bildbereiche segmentiert. Danach wird jeder Bereich durch ein geeignetes Komprimierungsverfahren codiert, mit dem eine möglichst hohe Kompressionsrate bei minimalem Verlust an Informationen, oder gar ohne Verlust an Informationen, erzielt werden kann.

Die zweite Gruppe beinhaltet Algorithmen, die in Abhängigkeit von lokalen Eigenschaften des Bildes ihre Parameter ändern, z.B. das Code-Buch bei der Vektor-Quantisierung oder die Parameter der Quantisierungsmatrix bei dem JPEG-Verfahren oder bei der Huffman-Codierung. Das digitale Bild wird dann durch ein Komprimierungsverfahren mit lokal variierenden Parametern codiert.

Die dritte Gruppe beinhaltet Algorithmen, die globale Eigenschaften des Bildes ermitteln und auf der Basis dieser globalen Eigenschaften das effektivste Komprimierungsverfahren auswählen. Danach wird das gesamte digitale Bild durch das ausgewählte Komprimierungsverfahren codiert.

All diesen aus dem Stand der Technik bekannten adaptiven Komprimierungsverfahren ist gemein, dass das zu komprimierende digitale Bild in mehrere lokale Bildbereiche segmentiert wird und die verschiedenen Komprimierungsverfahren dann auf die einzelnen Bildbereiche angewandt werden.

Es ist die Aufgabe der vorliegenden Erfindung, ein neuartiges Verfahren zum Komprimieren eines digitalen Bildes zu schaffen, das für beliebig ausgestaltete digitale Bilder eine möglichst effiziente Komprimierung ermöglicht, d.h. mit möglichst wenig Rechenaufwand eine möglichst hohe Kompressionsrate erzielen kann.

Zur Lösung dieser Aufgabe schlägt die Erfindung ausgehend von dem Verfahren der eingangs genannten Art ein Verfahren vor, das gekennzeichnet ist durch die nachfolgenden Schritte:

- Analysieren der Bitebenen des digitalen Bildes;
- Unterteilen der Bitebenen des digitalen Bildes in mehrere Bitebenenbereiche, die jeweils mindestens eine Bitebene aufweisen;
- Aufteilen des digitalen Bildes in mehrere Bitebenenbereichsbilder die jeweils einen der Bitebenenbereiche umfassen;

- Auswählen eines bestimmten Komprimierungsverfahrens für jedes Bitebenenbereichsbild; und
- Komprimieren der einzelnen Bitebenenbereichsbilder mit dem jeweils ausgewählten Komprimierungsverfahren.

Das zu komprimierende digitale Bild wird nicht, wie aus dem Stand der Technik bekannt, in mehrere Bildbereiche segmentiert, die jeweils dieselben Bitebenen wie das digitale Bild aufweisen. Es werden also nicht die Inhalte des digitalen Bildes, sondern dessen Bitebenen analysiert.

Die Bitebenen des digitalen Bildes werden in Abhängigkeit von dem Analyseergebnis in mehrere Bitebenenbereiche unterteilt. Das digitale Bild wird in mehrere Bitebenenbereichsbilder aufgeteilt, die jeweils einen der Bitebenenbereiche umfassen. Für jedes Bitebenenbereichsbild wird ebenfalls in Abhängigkeit von dem Ergebnis der Analyse der Bitebenen des digitalen Bildes ein bestimmtes Komprimierungsverfahren ausgewählt.

Die Bitebenen des digitalen Bildes können bspw. unter dem Gesichtspunkte einer möglichst hohen Komprimierungsrate, eines möglichst geringen Verlusts an Bildinhalten bei dem rekonstruierten Bild oder eines möglichst geringen Rechenaufwands analysiert werden. Die verwendeten Komprimierungsverfahren werden vorzugsweise aus den in einem Browser üblicherweise enthaltenen herkömmlichen Komprimierungsverfahren ausgewählt. Schließlich werden die einzelnen Bitebenenbereichsbilder mit dem jeweils ausgewählten Komprimierungsverfahren komprimiert.

Anhand der Analyse der Bitebenen des zu komprimierenden digitalen Bildes werden somit die Bitebenen in bestimmte Bitebenenbereiche unterteilt, das digitale Bild in bestimmte Bitebenenbereichsbilder aufgeteilt und für jedes Bitebenenbereichsbild ein geeignetes Komprimierungsverfahren ausgewählt. Das erfindungsgemäße Verfahren zum Komprimieren eines digitalen Bildes ermöglicht eine besonders effiziente

Komprimierung des digitalen Bildes. Mit einem vergleichsweise geringen Rechenaufwand können besonders hohe Komprimierungsraten mit einem äußerst geringen oder sogar keinem Verlust an Bildinhalten erzielt werden.

Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass die Bildebenen des digitalen Bildes in zwei Bitebenenbereiche unterteilt werden, das digitale Bild in zwei Bitebenenbereichsbilder aufgeteilt wird, für den ersten Bitebenenbereich, der die höherwertigen Bitebenen umfasst, das GIF-Komprimierungsverfahren und für den zweiten Bitebenenbereich, der die niederwertigen Bitebenen umfasst, das JPEG-Komprimierungsverfahren ausgewählt wird, und die Bitebenenbereichsbilder mit dem jeweils ausgewählten Komprimierungsverfahren komprimiert werden. Das GIF- und das JPEG-Verfahren sind zwei Komprimierungsverfahren, die sich wie eingangs bereits erläutert besonders gut ergänzen. Wo das eine Komprimierungsverfahren Nachteile hat, arbeitet das andere Komprimierungsverfahren besonders gut und umgekehrt. Durch den Einsatz dieser beiden Komprimierungsverfahren wird eine besonders effiziente Komprimierung eines digitalen Bildes ermöglicht. Zudem handelt es sich bei dem GIF- und dem JPEG-Komprimierungsverfahren um herkömmliche Komprimierungsverfahren, die in nahezu jedem modernen Browser enthalten sind und dort jederzeit aufgerufen werden können.

Durch die Analyse der Bitebenen des digitalen Bildes und die Aufteilung des digitalen Bildes in Bitebenenbereichsbilder wird eine besonders effiziente Aufteilung des digitalen Bildes in Bereiche, die mit dem GIF-Komprimierungsverfahren komprimiert werden, und in Bereiche, die mit dem JPEG-Komprimierungsverfahren komprimiert werden, möglich. Eine solche effiziente Aufteilung des digitalen Bildes könnte durch eine bloße Analyse der Bildinhalte des digitalen Bildes nicht erreicht werden. In den höherwertigen Bitebenen sind hauptsächlich die Bildinhalte des digitalen Bildes und in den niederwertigen Bitebenen ist hauptsächlich das Rauschen des

digitalen Bildes enthalten. Die höherwertigen Bitebenen werden mit dem GIF-Verfahren und die niederwertigen Bitebenen einschließlich des Rauschens mit dem JPEG-Verfahren komprimiert.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass das Analysieren der Bitebenen des digitalen Bildes die nachfolgenden Schritte umfasst:

- Aufteilen des digitalen Bildes in mehrere Bitebenenbilder, die jeweils eine Bitebene umfassen;
- Generieren von Analysebildern durch Überlagern mehrerer Bitebenenbilder, beginnend bei den Bitebenenbildern, die die zwei höchstwertigen Bitebenen umfassen, für das erste Analysebild und durch Hinzunahme des Bitebenenbildes, das die jeweils nächst niedrigere Bitebene umfasst, für jedes weitere Analysebild;
- Segmentieren der Analysebilder in Segmente, die jeweils dieselben Pixelwerte aufweisen;
- Ermitteln der Flächen der Segmente;
- Bilden des Mittelwertes der Segmentflächen eines Analysenbildes;
- Ermitteln des Kompressionsfaktors in Abhängigkeit von dem Mittelwert der Segmentflächen beim Einsatz eines bestimmten Komprimierungsverfahrens;
- Ermitteln des Analysebildes, bei dem der Kompressionsfaktor beim Einsatz eines bestimmten Komprimierungsverfahrens größer als ein vorgegebener Grenzwertfaktor ist; und
- Bestimmen der Bitebenenbereiche in Abhängigkeit von denjenigen Bitebenen, die das ermittelte Analysebild umfasst.

Vorteilhafterweise wird ausgehend von dem Analysebild, das die Bitebenenbilder mit den höchstwertigen Bitebenen umfasst, dasjenige Analysebild ermittelt, bei dem der Kompressionsfaktor beim Einsatz eines bestimmten

Komprimierungsverfahrens größer als 10 ist. Dies gilt insbesondere für das LZW-Verfahren, nach dem die digitalen Bilder in dem GIF-Format komprimiert werden.

Die einzelnen Analysebilder werden in Segmente segmentiert, die jeweils dieselben Pixelwerte, d. h. dieselben Farben oder Graustufen aufweisen. Falls die Flächen der Segmente eines Analysebildes relativ großflächig sind, eignet sich ein Komprimierungsverfahren, das auf großflächige digitale Bilder derselben Farbe oder Graustufe ausgelegt ist, wie bspw. das GIF-Verfahren, wesentlich besser zum Komprimieren dieses Analysebildes als ein Komprimierungsverfahren, das auf hochauflösende digitale Bilder mit hoher Feindetaillierung und einer Vielzahl unterschiedlicher Farben und Graustufen ausgelegt ist, wie bspw. das JPEG-Verfahren.

Das Verfahren gemäß der vorliegenden Ausführungsform ermöglicht eine Aufteilung des digitalen Bildes in mehrere Bitebenenbereichsbilder, die genau diejenigen Bitebenen des digitalen Bildes umfassen, dass sie beim Einsatz eines geeigneten Komprimierungsverfahrens mit einem Kompressionsfaktor komprimiert werden, der größer als ein vorgegebener Grenzwertfaktor, vorzugsweise größer als 10, ist. Da der Kompressionsfaktor bestimmter Komprimierungsverfahren, bspw. des GIF-Komprimierungsverfahrens, abhängig ist von dem Mittelwert der Segmentflächen der Analysebilder, kann der Kompressionsfaktor bei der vorliegenden Ausführungsform auf einfache Weise schnell und zuverlässig bestimmt werden. Vorzugsweise werden die Bitebenen des digitalen Bildes derart in Bitebenenbereiche unterteilt, dass bei Einsatz eines bestimmten Komprimierungsverfahrens ein Kompressionsfaktor größer 10 erzielt wird. Erzielen mehrere Komprimierungsverfahren einen Kompressionsfaktor größer 10, so wird dasjenige Verfahren ausgewählt, das den geringsten Rechenaufwand oder den höchsten Kompressionsfaktor aufweist.

Gemäß einer weiteren bevorzugten Ausführungsform der

vorliegenden Erfindung wird vorgeschlagen, dass vor dem Komprimieren des ersten Bitebenenbereichsbildes, das die höherwertigen Bitebenen umfasst, die Richtung des GIF-Komprimierungsverfahrens ermittelt wird, mit der die höchste Kompressionsrate erzielt werden kann. Die Richtung des GIF-Komprimierungsverfahrens wird vorzugsweise mit Hilfe der nachfolgenden Schritte ermittelt:

- Aufteilen des digitalen Bildes in mehrere Bitebenenbilder, die jeweils eine Bitebene umfassen; und
- Generieren von Analysebildern durch Überlagern mehrerer Bitebenenbilder, beginnend bei den Bitebenenbildern, die die zwei höchstwertigen Bitebenen umfassen, für das erste Analysebild und durch Hinzunahme des Bitebenenbildes, das die jeweils nächst niedrigere Bitebene umfasst, für jedes weitere Analysebild;
- Segmentieren der Analysebilder in Segmente, die jeweils dieselben Pixelwerte aufweisen;
- Ermitteln der vertikalen Seitenlängen und der horizontalen Seitenlängen der Segmente;
- Bilden des Mittelwerts der vertikalen Seitenlängen und der horizontalen Seitenlängen eines Analysebildes;
- Vergleich des Mittelwerts der vertikalen Seitenlängen mit dem Mittelwert der horizontalen Seitenlängen eines Analysebildes; und
- Ermitteln der Richtung des GIF-Komprimierungsverfahrens aus dem Ergebnis des Vergleichs der Mittelwerte der Seitenlängen.

Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass das zweite Bitebenenbereichsbild, das die niederwertigen Bitebenen umfasst, vor dem Komprimieren in einen Objektbereich und in einen Hintergrundbereich klassifiziert wird. Der Objektbereich umfasst ein oder mehrere in dem digitalen Bild darzustellende Objekte. Der Hintergrundbereich ist unabhängig von dem oder jedem darzustellenden Objekt und hat keinen Einfluss auf die Darstellung des Objekts. Ohne Verlust an Qualität des

rekonstruierten Bildes kann der Kompressionsfaktor des Komprimierungsverfahrens erhöht werden, wenn der Hintergrund vereinfacht wird, d. h. eine geringere Auflösung und/oder weniger Farben oder Graustufen aufweist.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass die Klassifizierung des zweiten Bitebenenbereichsbildes mit Hilfe der nachfolgenden Schritte durchgeführt wird:

- Segmentieren des ersten Bitebenenbereichsbildes, das die höherwertigen Bitebenen umfasst, in eine Vielzahl von Pixelblöcken;
- Segmentieren der Pixelblöcke in Segmente, die jeweils dieselben Pixelwerte aufweisen;
- Ermitteln der Anzahl der verschiedenen Segmente mit unterschiedlichen Pixelwerten innerhalb eines Pixelblocks;
- Klassifizieren eines Pixelblocks als Objektbereich, falls in dem Pixelblock verschiedene Segmente enthalten sind;
- sonst Klassifizieren des Pixelblocks als Hintergrundbereich; und
- Klassifizieren des zweiten Bitebenenbereichsbildes, das die niederwertigen Bitebenen umfasst, in entsprechenden Objektbereich und Hintergrundbereich.

Gemäß einer anderen bevorzugten Ausführungsform der vorliegenden Erfindung wird der Hintergrundbereich des zweiten Bitebenenbereichsbildes, das die niederwertigen Bitebenen umfasst, durch einen Hintergrund bestehend aus quadratischen Pixelblöcken ersetzt, wobei alle Pixel eines Pixelblocks denselben Pixelwert aufweisen. Die Werte aller Pixel eines Pixelblocks sind vorteilhafterweise gleich dem Mittelwert der Pixelwerte des Pixelblocks. Vorzugsweise wird der Hintergrundbereich durch einen Hintergrund bestehend aus Pixelblöcken mit einer Größe von 8x8 Pixel ersetzt. Durch diese Maßnahmen zur Vereinfachung des Hintergrundbereichs kann der Kompressionsfaktor des JPEG-Komprimierungsverfahrens

weiter erhöht werden, ohne dass es zu einem Verlust an Qualität der Bildinformationen des rekonstruierten Bildes kommt.

Gemäß einer anderen vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass vor dem Komprimieren des zweiten Bitebenenbereichsbildes, das die niederwertigen Bitebenen umfasst, der Parameter Q für das JPEG-Komprimierungsverfahren ermittelt wird.

Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass der Parameter Q in Abhängigkeit von der Anzahl der Bitebenenbilder des zweiten Bitebenenbereichsbildes, die Bildinformationen enthalten ermittelt wird.

Es wird des Weiteren vorgeschlagen, dass der Parameter Q vorteilhafterweise an Hand der nachfolgenden Schritte ermittelt wird:

- Aufteilen des digitalen Bildes in mehrere Bitebenenbilder, die jeweils eine Bitebene des digitalen Bildes umfassen;
- Segmentieren der Bitebenenbilder des zweiten Bitebenenbereichsbildes, das die niederwertigen Bitebenen umfasst, in Segmente, die jeweils dieselben Pixelwerte aufweisen;
- Ermitteln der Flächen der Segmente;
- Bilden des Mittelwerts der Segmentflächen eines Bitebenenbildes; und
- ausgehend von dem höchstwertigsten Bitebenenbild des zweiten Bitebenenbereichsbildes, Ermitteln der Anzahl der Bitebenenbilder, bei denen die Differenz des Mittelwerts der Segmentflächen dieses Bitebenenbildes und des Mittelwerts der Segmentflächen des niederwertigsten Bitebenenbildes größer als 10% des Mittelwerts der Segmentflächen des niederwertigsten Bitebenenbildes ist.

Die Abhängigkeit zwischen der ermittelten Anzahl der Bitebenenbilder und dem Parameter Q wird durch eine statische Modellierung für jede JPEG-Version berechnet.

Gemäß einer anderen vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass die mit verschiedenen Komprimierungsverfahren komprimierten Bitebenenbereichsbilder zu einer komprimierten Bilddatei zusammengefasst werden, in deren Kopfzeile Informationen über die Größe der einzelnen Bitebenenbereichsbilder geschrieben werden. Die Kopfzeile enthält also Informationen darüber, wo bei der Komprimierung die Grenzen zwischen den einzelnen Bitbereichen gezogen wurden.

Vorteilhafterweise werden in die Kopfzeile der Bilddatei auch Informationen darüber geschrieben, welches Bitebenenbereichsbild mit welchem Komprimierungsverfahren komprimiert wurde. Anhand der in die Kopfzeile geschriebenen Informationen kann eine komprimierte Bilddatei zu einem späteren Zeitpunkt problemlos wieder in die komprimierten Bitebenenbereichsbilder zerlegt und die einzelnen Bitebenenbereichsbilder mit dem entsprechenden Dekomprimierungsverfahren dekomprimiert werden.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, ein Verfahren zum Übertragen eines digitalen Bildes der eingangs genannten Art zu schaffen, das bei beliebig ausgestalteten digitalen Bildern, d.h. bei Fotos ebenso wie bei Zeichnungen, eine möglichst schnelle Übertragung des digitalen Bildes ermöglicht.

Zur Lösung dieser Aufgabe schlägt die Erfindung ausgehend von dem Verfahren zum Übertragen eines digitalen Bildes der eingangs genannten Art ein Verfahren vor, das gekennzeichnet ist durch die nachfolgenden Schritte:

- Komprimieren des digitalen Bildes auf dem Computer des Senders mit Hilfe eines Verfahrens nach Anspruch 15 oder

16;

- Übermitteln der komprimierten Bilddatei von dem Endgerät des Senders zu dem Endgerät des Empfängers;
- Empfangen der komprimierten Bilddatei auf dem Endgerät des Empfängers;
- Analysieren der Kopfzeile der komprimierten Bilddatei auf dem Endgerät des Empfängers;
- Zerlegen der komprimierten Bilddatei in die komprimierten Bitebenenbereichsbilder;
- Dekomprimieren der Bitebenenbereichsbilder mit den entsprechenden Dekomprimierungsverfahren auf dem Endgerät des Empfängers; und
- Überlagern der dekomprimierten Bitebenenbereichsbilder zu dem digitalen Bild auf dem Endgerät des Empfängers.

Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass mit der komprimierten Bilddatei auch ein auf dem Endgerät des Empfängers lauffähiges Softwareprogramm übermittelt wird, das die Analyse der Kopfzeile der komprimierten Bilddatei ausführt, das Zerlegen der komprimierten Bilddatei und das Dekomprimieren der komprimierten Bitebenenbereichsbilder steuert und das Überlagern der dekomprimierten Bitebenenbereichsbilder zu dem digitalen Bild ausführt.

Das Softwareprogramm ist vorteilhafter Weise als ein Java-Applet ausgebildet, das in einem Browser des Endgeräts des Empfängers lauffähig ist.

Ein bevorzugtes Ausführungsbeispiel der vorliegenden Erfindung wird im Folgenden anhand der Zeichnungen näher erläutert. Es zeigen:

Fig. 1 ein Ablaufdiagramm des erfindungsgemäßen Verfahren zum Komprimieren eines digitalen Bildes gemäß einer bevorzugten Ausführungsform;

Fig. 2 das Aufteilen des digitalen Bildes in mehrere Analysebilder;

Fig. 3 das Segmentieren der Analysebilder; und

Fig. 4 das Aufteilen des digitalen Bildes in zwei Bitebenenbereichsbilder.

Bei dem erfindungsgemäßen Verfahren handelt es sich um ein adaptives Komprimierungsverfahren für digitale Bilder, deren Bildinformationen in mehreren Bitebenen codiert sind. Anders als bei den aus dem Stand der Technik bekannten Komprimierungsverfahren werden nicht die Inhalte des digitalen Bildes analysiert, wird also das zu komprimierende digitale Bild nicht in mehrere Bildbereiche segmentiert, die jeweils dieselben Bitebenen wie das digitale Bild aufweisen. Vielmehr wird das digitale Bild bei dem erfindungsgemäßen Verfahren anhand seiner Bitebenen analysiert, und dann werden in Abhängigkeit von dem Analyseergebnis geeignete Komprimierungsverfahren für bestimmte Bitebenen ausgewählt.

Ein Ablaufdiagramm des erfindungsgemäßen Verfahrens ist in Fig. 1 dargestellt. Bei dem zu komprimierende digitalen Bild DB (Block 1) handelt es sich um die Aufnahme eines Computerdiagnostikgeräts aus der Medizintechnik. Das digitale Bild DB weist zwölf Bitebenen BE auf, in denen die Bildinformationen codiert sind, d. h. jedes Pixel des digitalen Bildes DB hat 12-bit. Die anschließend in einem Block 2 ausgeführten Vorgänge sind zur Erläuterung in Fig. 2 detailliert dargestellt. In Block 2 wird das digitale Bild DB in zwölf Bitebenenbilder BEB aufgeteilt, die jeweils eine Bitebene BE umfassen. Dann werden aus den Bitebenenbildern BEB Analysebilder AB generiert. Zum Generieren des ersten Analysebildes AB1 werden die beiden Bitebenenbilder BEB1 und BEB2 mit den höchstwertigen Bitebenen BE1 und BE2 überlagert. Für das nächste Analysebild AB2 wird dem ersten Analysebild AB1 das Bitebenenbild BEB3 mit der nächst niedrigeren Bitebene

BE3 überlagert. Zum Generieren der übrigen Analysebilder AB3 bis AB11 wird ebenso verfahren, so dass schließlich aus den zwölf Bitebenenbilder BEB elf Analysebilder AB generiert werden.

Anschließend werden in Block 3 die Analysebilder in Segmente segmentiert, die jeweils dieselben Pixelwerte, d. h. dieselbe Farbe oder Graustufe, aufweisen. Dieser Vorgang ist zur Erläuterung in Fig. 3 genauer dargestellt. In Fig. 3 ist links ein Analysebild AB beispielhaft herausgegriffen und ein Bereich des Analysebildes AB rechts vergrößert dargestellt. Die einzelnen Segmente in diesem Bereich des Analysebildes AB sind mit S1 bis S9 gekennzeichnet. Dann werden, noch immer in Block 3, die Flächen der einzelnen Segmente ermittelt. Das Segment S1 hat bspw. eine Fläche von 7 und das Segment S8 eine Fläche von 34. Schließlich werden die Mittelwerte der Segmentflächen eines jeden Analysebildes AB gebildet.

Zwischen dem Mittelwert der Segmentflächen und dem Kompressionsfaktor des GIF-Komprimierungsverfahrens besteht eine Abhängigkeit. Aufgrund dieser Abhängigkeit wird in den Blöcken 4 bis 6 beginnend bei der höchstwertigsten Bitebene die Anzahl n der Bitebenen BE berechnet, bei denen die Bedingung, dass der Kompressionsfaktor bei der Kompression eines Analysebildes AB mit einem bestimmten Kompressionsfaktor größer als 10 ist, noch richtig ist.

Die Blöcke 4 bis 6 liefern einen Grenzwert, an dem die Bitebenen BE des digitalen Bildes DB in einen niederwertigen Bitebenenbereich und in einen höherwertigen Bitebenenbereich unterteilt werden. Das digitale Bild DB wird dann in ein erstes Bitebenenbereichsbild BEBB1, das den Bitebenenbereich mit den höherwertigen Bitebenen BE1 bis BE4 umfasst, (Block 7) und in ein zweites Bitebenenbereichsbild BEBB2, das den Bitebenenbereich mit den niederwertigen Bitebenen BE5 bis BE12 umfasst, aufgeteilt (Block 8). Das erste Bitebenenbereichsbild BEBB1 entspricht dem Analysebild A3, das die Bitebenenbilder

BEB1 bis BEB4 mit den entsprechenden Bitebenen BE1 bis BE4 umfasst. Das zweite Bitebenenbereichsbild BEBB2 entspricht der Differenz des digitalen Bildes DB und dem ersten Bitebenenbereichsbild BEBB1. Die Aufteilung des digitalen Bildes DB in das erste und das zweite Bitebenenbereichsbild BEBB1 bzw. BEBB2 ist in Fig. 4 verdeutlicht. Für das erste Bitebenenbereichsbild BEBB1 wird das GIF-Komprimierungsverfahren ausgewählt und für das zweite Bitebenenbereichsbild BEBB2 das JPEG-Komprimierungsverfahren.

In Block 9 wird die Richtung des GIF-Komprimierungsverfahrens ermittelt, mit der die höchste Kompressionsrate erzielt werden kann. Dazu werden die vertikalen Seitenlängen und die horizontalen Seitenlängen der Segmente der Analysebilder AB ermittelt (vgl. Fig. 3). Das Segment S1 hat bspw. eine vertikale Seitenlänge (Höhe) von 2 und eine horizontale Seitenlänge (Breite) von 4 und das Segment S8 eine Höhe von 4 und eine Breite von 12. Dann wird der Mittelwert der vertikalen Seitenlängen und der horizontalen Seitenlängen eines jeden Analysebildes AB gebildet. Aus einem Vergleich des Mittelwerts der vertikalen Seitenlängen mit dem Mittelwert der horizontalen Seitenlängen eines jeden Analysebildes AB wird schließlich die Richtung des GIF-Komprimierungsverfahrens gebildet.

In den Blöcken 10 bis 12 wird die Anzahl k der Bitebenen BE des zweiten Bitebenenbereichsbildes BEBB2 ermittelt, die Bildinformationen enthalten. Es wird davon ausgegangen, dass eine Bitebene BE dann Bildinformationen enthält, wenn das Verhältnis Peak-Signal-to-Noise-Ratio (PSNR) größer als 40 dB ist, d. h. wenn die Differenz zwischen dem Mittelwert der Segmentflächen dieses Bitebenenbildes BEB und dem Mittelwert der Segmentfläche des niederwertigsten Bitebenenbildes BEB12 kleiner als 10% des Mittelwerts der Segmentflächen des niederwertigsten Bitebenenbildes BEB12 ist. Ausgehend von der höchstwertigen Bitebene des zweiten Bitebenenbereichsbildes wird dies in dem Block 11 überprüft.

In Block 13 wird der Parameter Q für das JPEG-Verfahren ermittelt. Der Parameter Q wird in Abhängigkeit von der Anzahl k der Bitebenen BE5 bis BE12 des zweiten Bitebenenbereichsbildes BEBB2 ermittelt, die Bildinformationen enthalten.

Anschließend wird das zweite Bitebenenbereichsbild BEBB2 in Block 14 in einen Objektbereich und in einen Hintergrundbereich klassifiziert. Dazu wird das erste Bitebenenbereichsbild BEBB1 in eine Vielzahl von Pixelblöcken segmentiert. Die Pixelblöcke werden ihrerseits in Segmente, die jeweils dieselben Pixelwerte aufweisen, segmentiert. Dann wird die Anzahl der verschiedenen Segmente mit unterschiedlichen Pixelwerten innerhalb jeweils eines Pixelblocks ermittelt. Ein Pixelblock wird als Objektbereich klassifiziert, falls in dem Pixelblock verschiedene Segmente mit zwei oder mehr unterschiedlichen Pixelwerten enthalten sind. Sonst wird der Pixelblock als Hintergrundbereich klassifiziert. Das zweite Bitebenenbereichsbild BEBB2 wird dann in einen entsprechenden Objektbereich und Hintergrundbereich klassifiziert.

Zur Erhöhung der Kompressionsrate wird der Hintergrundbereich durch einen Hintergrund bestehend aus quadratischen Pixelblöcken mit einer Größe von 8x8 Pixel ersetzt. Alle Pixel eines Pixelblocks weisen denselben Pixelwert auf, nämlich den Mittelwert der Pixelwerte des Pixelblocks.

In Block 15 wird das erste Bitebenenbereichsbild BEBB1 in der in Block 9 ermittelten Richtung mit dem GIF-Komprimierungsverfahren komprimiert. Ebenso wird in Block 16 das zweite Bitebenenbereichsbild BEBB2 mit dem in Block 13 ermittelten Parameter Q und dem in Block 14 vereinfachten Hintergrundbereich mit dem JPEG-Komprimierungsverfahren komprimiert. Die komprimierten Bitebenenbereichsbilder werden in Block 17 zu einer komprimierten Bilddatei (Block 18) zusammengefaßt. In einer Kopfzeile der komprimierten Bilddatei

werden Informationen über die Größe der einzelnen Bitebenenbereichsbilder und über die auf die einzelnen Bitebenenbereichsbilder BEBB angewandten Komprimierungsverfahren geschrieben.

Patentansprüche

1. Verfahren zum Komprimieren eines digitalen Bildes (DB), dessen Bildinformationen in mehreren Bitebenen (BE) codiert sind, **gekennzeichnet durch** die nachfolgenden Schritte:
 - Analysieren der Bitebenen (BE) des digitalen Bildes (DB);
 - Unterteilen der Bitebenen (BE) des digitalen Bildes (DB) in mehrere Bitebenenbereiche, die jeweils mindestens eine Bitebene (BE) aufweisen;
 - Aufteilen des digitalen Bildes (DB) in mehrere Bitebenenbereichsbilder (BEBB), die jeweils einen der Bitebenenbereiche umfassen;
 - Auswählen eines bestimmten Komprimierungsverfahrens für jedes Bitebenenbereichsbild (BEBB); und
 - Komprimieren der einzelnen Bitebenenbereichsbilder (BEBB) mit dem jeweils ausgewählten Komprimierungsverfahren.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Bildebenen (BE) des digitalen Bildes (DB) in zwei Bitebenenbereiche unterteilt werden, das digitale Bild in zwei Bitebenenbereichsbilder (BEBB1; BEBB2) aufgeteilt wird, für den ersten Bitebenenbereich, der die höherwertigen Bitebenen (BE) umfasst, das GIF-Komprimierungsverfahren und für den zweiten Bitebenenbereich, der die niederwertigen Bitebenen (BE) umfasst, das JPEG-Komprimierungsverfahren ausgewählt wird, und die Bitebenenbereichsbilder (BEBB1; BEBB2) mit dem jeweils ausgewählten Komprimierungsverfahren komprimiert werden.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass das Analysieren der Bitebenen (BE) des digitalen Bildes (DB) die nachfolgenden Schritte umfasst:

- Aufteilen des digitalen Bildes (DB) in mehrere Bitebenenbilder (BEB), die jeweils eine Bitebene (BE) umfassen;
 - Generieren von Analysebildern (AB) durch Überlagern mehrerer Bitebenenbilder (BEB), beginnend bei den Bitebenenbildern (BEB1, BEB2), die die zwei höchstwertigen Bitebenen (BE) umfassen, für das erste Analysebild (AB1) und durch Hinzunahme des Bitebenenbildes (BEB3...BEB12), das die jeweils nächst niedrigere Bitebene (BE3...BE12) umfasst, für jedes weitere Analysebild (AB2...AB11);
 - Segmentieren der Analysebilder (AB) in Segmente (S), die jeweils dieselben Pixelwerte aufweisen;
 - Ermitteln der Flächen der Segmente (S);
 - Bilden des Mittelwertes der Segmentflächen eines Analysenbildes (AB);
 - Ermitteln des Kompressionsfaktors in Abhängigkeit von dem Mittelwert der Segmentflächen beim Einsatz eines bestimmten Komprimierungsverfahrens;
 - Ermitteln des Analysebildes (AB), bei dem der Kompressionsfaktor beim Einsatz eines bestimmten Komprimierungsverfahrens größer als ein vorgegebener Grenzwertfaktor ist; und
 - Bestimmen der Bitebenenbereiche in Abhängigkeit von denjenigen Bitebenen (BE), die das ermittelte Analysebild (AB) umfasst.
4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass ausgehend von dem Analysebild (AB1), das die Bitebenenbilder (BEB) mit den höchstwertigen Bitebenen (BE) umfasst, das Analysebild (AB) ermittelt wird, bei dem der Kompressionsfaktor beim Einsatz eines bestimmten Komprimierungsverfahrens größer als 10 ist.
5. Verfahren nach Anspruch 2, dadurch gekennzeichnet, dass vor dem Komprimieren des ersten Bitebenenbereichsbildes (BEBB1), das die höherwertigen Bitebenen (BE) umfasst,

die Richtung des GIF-Komprimierungsverfahrens ermittelt wird, mit der die höchste Kompressionsrate erzielt werden kann.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass die Richtung des GIF-Komprimierungsverfahrens mit Hilfe der nachfolgenden Schritte ermittelt wird:
 - Aufteilen des digitalen Bildes (DB) in mehrere Bitebenenbilder (BEB), die jeweils eine Bitebene (BE) umfassen;
 - Generieren von Analysebildern (AB) durch Überlagern mehrerer Bitebenenbilder (BEB), beginnend bei den Bitebenenbildern (BEB1, BEB2), die die zwei höchstwertigen Bitebenen (BE1, BE2) umfassen, für das erste Analysebild (AB1) und durch Hinzunahme des Bitebenenbildes (BEB3...BEB12), das die jeweils nächst niedrigere Bitebene (BE3...BE12) umfasst, für jedes weitere Analysebild (AB2...AB11);
 - Segmentieren der Analysebilder (AB) in Segmente (S), die jeweils dieselben Pixelwerte aufweisen;
 - Ermitteln der vertikalen Seitenlängen und der horizontalen Seitenlängen der Segmente (S);
 - Bilden des Mittelwerts der vertikalen Seitenlängen und der horizontalen Seitenlängen eines Analysebildes (AB);
 - Vergleich des Mittelwerts der vertikalen Seitenlängen mit dem Mittelwert der horizontalen Seitenlängen eines jeden Analysebildes (AB); und
 - Ermitteln der Richtung des GIF-Komprimierungsverfahrens aus dem Ergebnis des Vergleichs der Mittelwerte der Seitenlängen.
7. Verfahren nach Anspruch 2, dadurch gekennzeichnet, dass das zweite Bitebenenbereichsbild (BEBB2), das die niederwertigen Bitebenen (BE) umfasst, vor dem Komprimieren in einen Objektbereich und in einen Hintergrundbereich klassifiziert wird.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass die Klassifizierung des zweiten Bitebenenbereichsbildes (BEBB2) mit Hilfe der nachfolgenden Schritte durchgeführt wird:
- Segmentieren des ersten Bitebenenbereichsbildes, das die höherwertigen Bitebenen umfasst, in eine Vielzahl von Pixelblöcken;
 - Segmentieren der Pixelblöcke in Segmente, die jeweils dieselben Pixelwerte aufweisen;
 - Ermitteln der Anzahl der verschiedenen Segmente mit unterschiedlichen Pixelwerten innerhalb eines Pixelblocks;
 - Klassifizieren eines Pixelblocks als Objektbereich, falls in dem Pixelblock verschiedene Segmente enthalten sind;
 - sonst Klassifizieren des Pixelblocks als Hintergrundbereich; und
 - Klassifizieren des zweiten Bitebenenbereichsbildes (BEBB2), das die niederwertigen Bitebenen (BE) umfasst, in entsprechenden Objektbereich und Hintergrundbereich.
9. Verfahren nach Anspruch 7 oder 8, dadurch gekennzeichnet, dass der Hintergrundbereich des zweiten Bitebenenbereichsbildes (BEBB2), das die niederwertigen Bitebenen (BE) umfasst, durch einen Hintergrund bestehend aus quadratischen Pixelblöcken ersetzt wird, wobei alle Pixel eines Pixelblocks denselben Pixelwert aufweisen.
10. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass die Pixelwerte aller Pixel eines Pixelblocks gleich dem Mittelwert der Pixelwerte des Pixelblocks sind.
11. Verfahren nach Anspruch 9 oder 10, dadurch gekennzeichnet, dass der Hintergrundbereich durch einen Hintergrund bestehend aus Pixelblöcken mit einer Größe von 8x8 Pixel ersetzt wird.

12. Verfahren nach Anspruch 2, dadurch gekennzeichnet, dass vor dem Komprimieren des zweiten Bitebenenbereichsbildes (BEBB2), das die niederwertigen Bitebenen (BE) umfasst, der Parameter Q für das JPEG-Komprimierungsverfahren ermittelt wird.
13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass der Parameter Q in Abhängigkeit von der Anzahl der Bitebenenbilder (BEB) des zweiten Bitebenenbereichsbildes (BEBB2), die Bildinformationen enthalten, ermittelt wird.
14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, dass der Parameter Q an Hand der nachfolgenden Schritte ermittelt wird:
 - Aufteilen des digitalen Bildes (DB) in mehrere Bitebenenbilder (BEB), die jeweils eine Bitebene (BE) des digitalen Bildes (DB) umfassen;
 - Segmentieren der Bitebenenbilder (BEB) des zweiten Bitebenenbereichsbildes (BEBB), das die niederwertigen Bitebenen umfasst, in Segmente (S), die jeweils dieselben Pixelwerte aufweisen;
 - Ermitteln der Flächen der Segmente (S);
 - Bilden des Mittelwerts der Segmentflächen eines Bitebenenbildes (BEB) des Bitebenenbereichsbildes (BEBB2); und
 - ausgehend von dem höchstwertigen Bitebenenbild (BEB) des zweiten Bitebenenbereichsbildes (BEBB2), Ermitteln der Anzahl der Bitebenenbilder (BEB), bei denen die Differenz aus dem Mittelwert der Segmentflächen dieses Bitebenenbildes (BEB) und dem Mittelwert der Segmentflächen des niederwertigsten Bitebenenbildes (BEB12) größer als 10% des Mittelwerts der Segmentflächen des niederwertigsten Bitebenenbildes (BEB12) ist.
15. Verfahren nach einem der Ansprüche 1 bis 14, dadurch gekennzeichnet, dass die mit verschiedenen

Komprimierungsverfahren komprimierten Bitebenenbereichsbilder (BEBB) zu einer komprimierten Bilddatei zusammengefasst werden, in deren Kopfzeile Informationen über die Größe der einzelnen Bitebenenbereichsbilder (BEBB) geschrieben werden.

16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, dass in die Kopfzeile der komprimierten Bilddatei Informationen darüber geschrieben werden, welches Bitebenenbereichsbild (BEBB) mit welchem Komprimierungsverfahren komprimiert wurde.
17. Verfahren zum Übertragen eines digitalen Bildes (DB), dessen Bildinformationen in mehreren Bitebenen (BE) codiert sind, von dem Endgerät eines Senders zu dem Endgerät eines Empfängers, **gekennzeichnet durch die nachfolgenden Schritte:**
 - Komprimieren des digitalen Bildes (DB) auf dem Endgerät des Senders mit Hilfe eines Verfahrens nach Anspruch 15 oder 16;
 - Übermitteln der komprimierten Bilddatei von dem Endgerät des Senders zu dem Endgerät des Empfängers;
 - Empfangen der komprimierten Bilddatei auf dem Endgerät des Empfängers;
 - Analysieren der Kopfzeile der komprimierten Bilddatei auf dem Endgerät des Empfängers;
 - Zerlegen der komprimierten Bilddatei in die komprimierten Bitebenenbereichsbilder;
 - Dekomprimieren der Bitebenenbereichsbilder mit den entsprechenden Dekomprimierungsverfahren auf dem Endgerät des Empfängers; und
 - Überlagern der dekomprimierten Bitebenenbereichsbilder (BEBB) zu dem digitalen Bild (DB) auf dem Endgerät des Empfängers.
18. Verfahren nach Anspruch 17, dadurch gekennzeichnet, dass mit der komprimierten Bilddatei auch ein auf dem Endgerät

des Empfängers lauffähiges Softwareprogramm übermittelt wird, das die Analyse der Kopfzeile der komprimierten Bilddatei ausführt, das Zerlegen der komprimierten Bilddatei und das Dekomprimieren der komprimierten Bitebenenbereichsbilder steuert und das Überlagern der dekomprimierten Bitebenenbereichsbilder zu dem digitalen Bild ausführt.

19. Verfahren nach Anspruch 18, dadurch gekennzeichnet, dass das Softwareprogramm als ein Java-Applet ausgebildet ist, das in einem Browser des Endgeräts des Empfängers lauffähig ist.

1. / 3

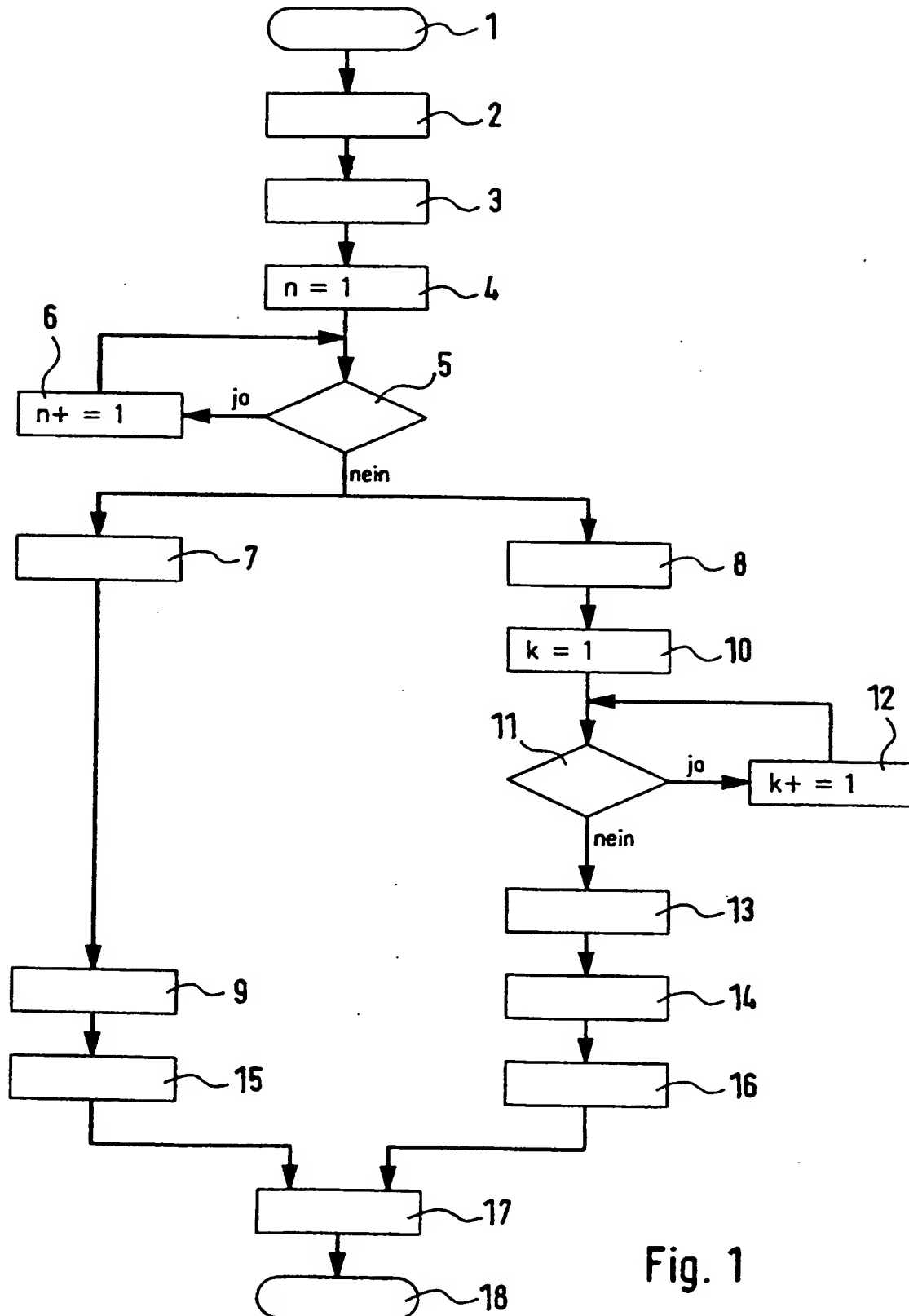
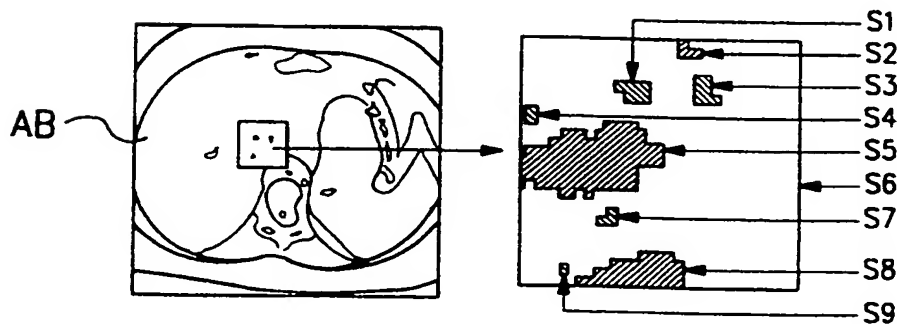
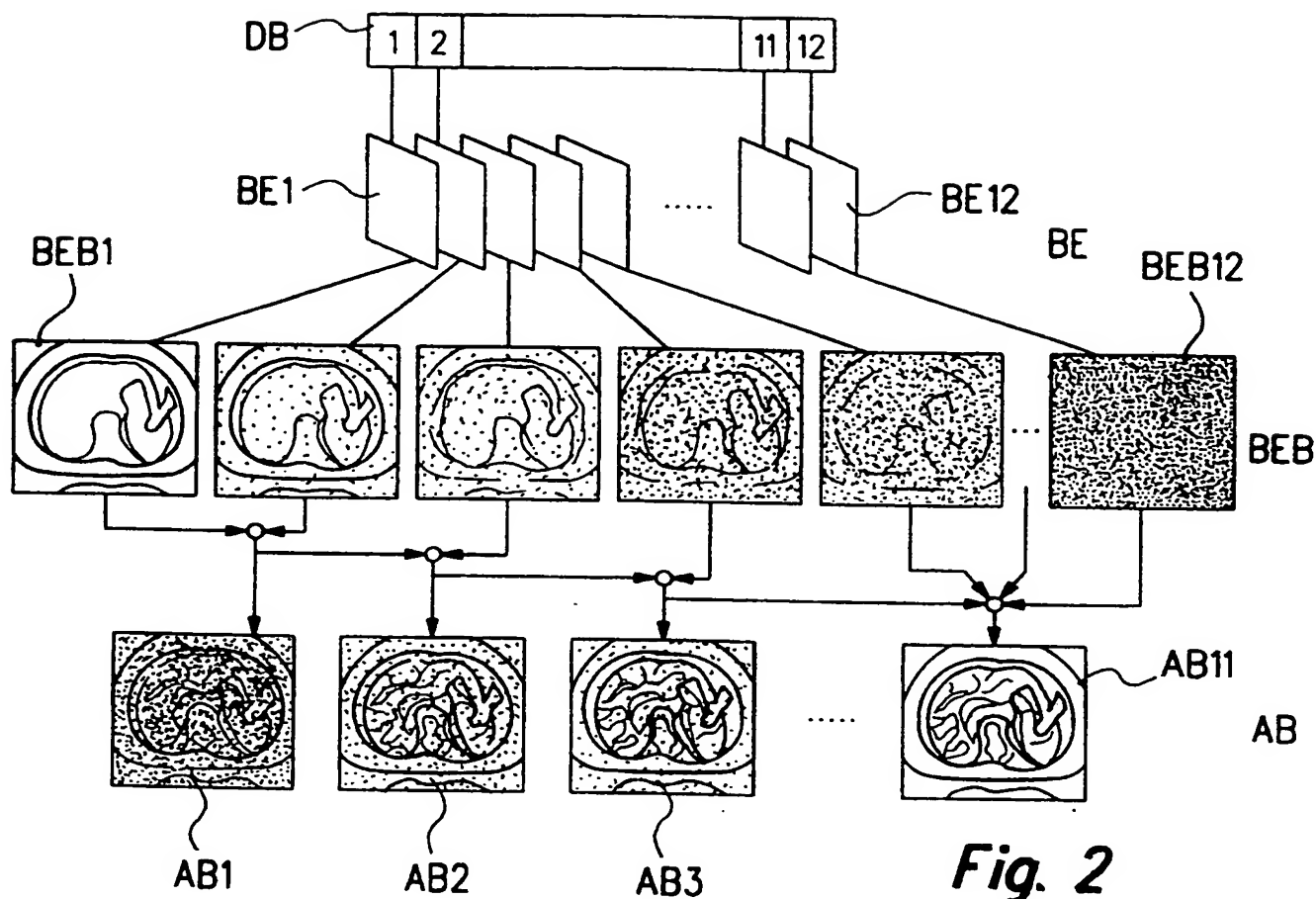


Fig. 1

THIS PAGE BLANK (USPTO)

2 / 3



THIS PAGE BLANK (USPTO)

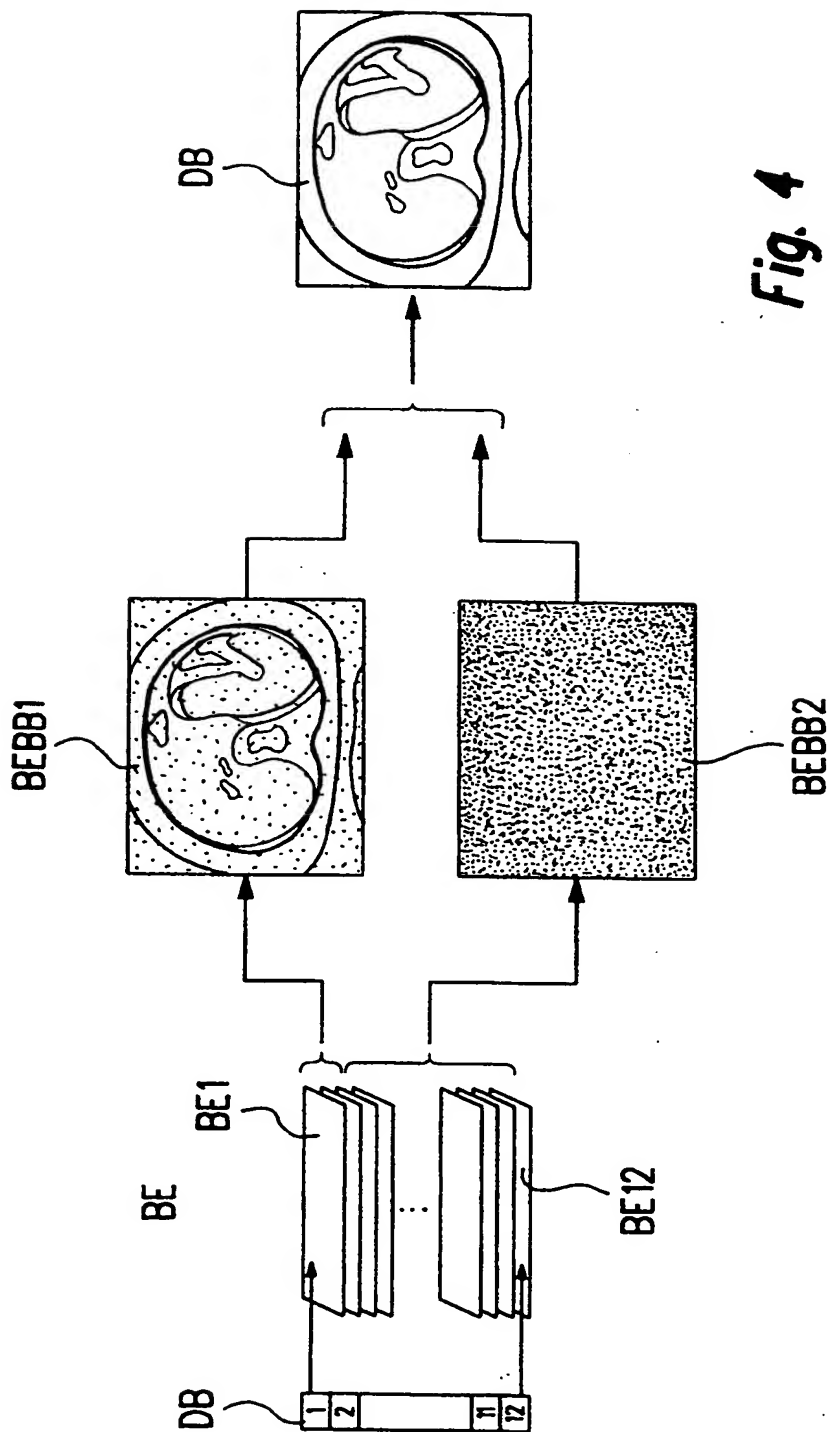


Fig. 4

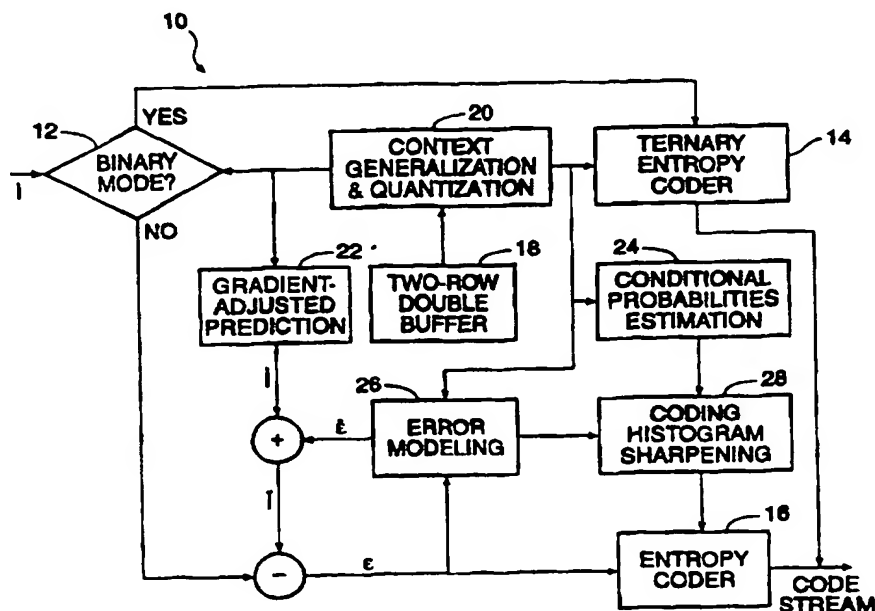
THIS PAGE BLANK (USPTO)



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04N 1/41		A1	(11) International Publication Number: WO 96/15620
			(43) International Publication Date: 23 May 1996 (23.05.96)
(21) International Application Number: PCT/CA95/00635 (22) International Filing Date: 8 November 1995 (08.11.95) (30) Priority Data: 9422738.6 10 November 1994 (10.11.94) GB (71) Applicant (for all designated States except US): THE CHINESE UNIVERSITY OF HONG KONG [GB/GB]; Research Administration Office, Room 226, Pi Chiu Building, Shatin, New Territories (HK). (71)(72) Applicant and Inventor: WU, Xiaolin [CA/CA]; 17; 683 Windemere Road, London, Ontario N5X 3T9 (CA). (72) Inventor; and (75) Inventor/Applicant (for US only): MEMON, Nasir [IN/US]; 128 Buena Vista Drive, Dekalb, IL 60115 (US). (74) Agent: MANNING, Gavin, N.; Oyen Wiggs Green & Mutala, 480 - 601 West Cordova Street, Vancouver, British Columbia V6B 1G1 (CA).			(81) Designated States: CA, JP, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: CONTEXT-BASED, ADAPTIVE, LOSSLESS IMAGE CODEC



(57) Abstract

An encoding/decoding method is provided for lossless (reversible) compression of digital pictures of all types, including continuous-tone images, graphics, multimedia images of mixed text, graphics and photographs, binary documents and drawings. Continuous-tone mode and binary mode are identified on a pixel-by-pixel basis (12). In continuous-tone mode, context modeling and prediction (20, 22, 26) are employed involving mostly integer arithmetic and simple logic in a conceptually sophisticated scheme. Both the encoding and decoding techniques are suitable for sequential and progressive transmission, although different specific algorithms may be employed for the different specific cases. The system is symmetric, meaning that the encoder and decoder have the same time and space complexities.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

CONTEXT-BASED, ADAPTIVE, LOSSLESS IMAGE CODEC

5

CLAIM OF PRIORITY

The present application claims partial priority of British Provisional Patent Application Serial No. 9422738-6 filed 10 November 1994.

10

BACKGROUND OF THE INVENTION

With rapidly-advancing computer, telecommunication, and digital imaging technologies, there is an astronomical amount of image data for a wide range of applications such as education, entertainment, medical imaging, space exploration, electronic publishing, visual arts, etc. This rapid growth of image data puts punishing burdens on computer storage and visual communication bandwidth. Thus image compression becomes a pressing technical challenge in visual communications and computing, without which it will be difficult to build, deploy, and use cost-effective multimedia information systems.

Lossless compression is a form of compression where an image can be reconstructed without any loss of information. Lossless image compression is required by medical imaging, satellite/aerial imaging, image archiving, preservation of precious art work and documents, the press, or any applications demanding ultra high image fidelity. Furthermore, lossless image coding is the necessary last step of many lossy image compression systems, such as lossless compression of codeword indices in vector quantization (VQ), and lossless compression of transform coefficients in Discrete Cosine Transform (DCT) and wavelet/subband-based coding.

There exists a large body of literature on lossless image compression algorithms and systems, such as the IBM Q-coder, and JPEG lossless coder. Among notable patents and publications are the US patents and research publications listed below:

SUBSTITUTE SHEET (RULE 26)

- 4,463,342 1984 IBM.
4,749,983 07/1988 Langdon.
4,969,204 11/1989 Melnychuck et al.
5,050,230 09/1990 Jones et al.
- 5 Universal Modeling and Coding -- J. Rissanen and G. Langdon, 1981, IEEE, vol. IT-27.
A Universal Data Compression System -- J. Rissanen, 1983, IEEE, vol. IT-29.
Parameter Reduction and Context Selection for Compression of
10 the Gray-Scale Images -- S. Todd, G. Langdon, and J. Rissanen, 1985, IBM J. Res. & Develop., vol. 29.
Comparing the Lossless Image Compression Standards and
Universal Context Modelling -- R. Arps, M. Weinberger, T.
Truong, and J. Rissanen, Proc. of the Picture Coding
15 Symposium, Sacramento, September 1994.
On the JPEG Model for Lossless Image Compression --
G. Langdon, A. Gulati, and E. Seiler,
Proc. of 1992 Data Compression Conf.
New Methods for lossless Image Compression Using Arithmetic
20 Coding --P. Howard and J. Vitter, 1992, Info. Proc. & Manag., vol. 28.

The currently achievable lossless compression ratio is still modest, being typically from 1.5:1 to 2.5:1. For instance, in contrast to the success of JPEG's lossy
25 compression standard, the current JPEG's lossless compression standard has sufficiently poor coding efficiency that it is seldom used in practice.

In 1994, ISO and JPEG solicited for proposals for the next international standard for lossless image
30 compression. The present invention is a result of the inventors' response to the ISO solicitation. The lead inventor Xiaolin Wu, developed a context-based, adaptive, lossless image coding and decoding technique (herein CALIC). Among nine proposals that were submitted to ISO for its
35 initial evaluation as candidates for the lossless image compression standard in 1995, the present CALIC system ranked first according to a criterion that accounts for both coding efficiency and algorithm simplicity.

Known prior art on lossless compression of continuous-tone images is based on the principle of predictive coding. An image is traversed, and pixels are encoded in a fixed order, typically in raster scan sequence. Previously encoded pixels that are known to both the encoder and the decoder are used to predict the upcoming pixels. The prediction errors rather than the pixels themselves are entropy encoded by Huffman or like arithmetic coding. The original image is reconstructed by adding the error term back to the prediction value. The predictive coding works because the histogram of the errors is much more concentrated (heavily biased toward 0) than the histogram of the pixel values, resulting in a significantly smaller zero-order entropy for the former than for the latter. Among numerous prediction schemes in the literature, the simplest type is a fixed linear predictor such as those used under the current lossless JPEG standard.

A linear predictor can be optimized on an image-by-image or even block-by-block basis via linear regression. However, such an optimization is expensive and brings only modest improvement in coding efficiency. Moreover the performance of linear predictors is not robust in the areas of edges. Adaptive, non-linear predictors can adjust parameters according to the local edge strengths and orientations, if edges exist. The adjustment of predictor parameters can be made very efficient since it is based on local information.

Historically, lossless image compression inherited the theoretical framework and methodology of text compression. Statistical modeling of the source being compressed plays a central role in any data compression systems. Suppose that we encode a finite source x_1, x_2, \dots, x_n sequentially. The optimal code length of the sequence in bits is then

$$-\log \prod_{i=0}^{n-1} p(x_{i+1} | x_1, \dots, x_i), \quad (1)$$

given the assignments of conditional probabilities. Arithmetic coding can approach this code length of the source.

The challenge is to assign the conditional probabilities $p(x_{i+1}|x_1, \dots, x_i)$ to maximize the product given above, hence minimize the code length. The achievable compression is governed by a scheme, or a model, that can assign high conditional probability distributions $p(x_{i+1}|x_1, \dots, x_i)$ to the given sequence.

Fitting a given source well with statistical models is a difficult and computationally very expensive task. Context modeling of continuous-tone images is made even more difficult by the large alphabet size (≥ 256) of grey-scale images. Context modeling of the source symbols (pixel values) would lead to an unwieldily large number of possible model states (contexts). This is more than a problem of high time and space complexities for modeling. If the number of model parameters is too large with respect to the image resolution, there will be insufficient samples to obtain good estimates of conditional probabilities on the model states, leading to poor coding efficiency. This is known as a "context dilution problem." This problem was theoretically formulated by Rissanen in the framework of stochastic complexity as the "model cost." Rissanen's work proves that the high complexity of a model can reduce coding efficiency, as observed by many data compression practitioners. What is needed are innovative algorithmic techniques to reduce the model complexity for improving both coding and computational efficiency.

SUMMARY OF THE INVENTION

According to the invention, an encoding/decoding method is provided for lossless (reversible) compression of digital pictures of all types, including continuous-tone images, graphics, multimedia images of mixed text, graphics and photographs, binary documents and drawings. Continuous-tone mode and binary mode are identified on a pixel-by-pixel basis. In continuous-tone mode, context modeling and prediction are employed, involving mostly integer arithmetic and simple logic in a conceptually sophisticated scheme. Both the encoding and decoding techniques are suitable for sequential and progressive transmission, although different

specific algorithms may be employed for the different specific cases. The system is symmetric, meaning that the encoder and decoder have the same time and space complexities.

5 A primary reason for the improved coding efficiency by context modeling of errors and error feedback lies in the fact that the prediction error sequence is a composition of multiple sources of distinct underlying statistics. The use of proper contexts to separate the statistically distinct sources of errors from the code stream can get a significant
10 reduction in conditional entropies.

The invention will be better understood by reference to the following detailed description in connection with the accompanying drawings.

15 BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic description of an encoding process for compression according to the invention in which the decoding process is a reverse.

20 Fig. 2 illustrates labeling of neighboring pixels for context-based prediction and modeling.

Fig. 3 shows four possible convexity relationships between a GAP predictor \hat{I} and two neighboring pixels.

25 Fig. 4 illustrates a two-stage adaptive prediction scheme via context modeling of prediction errors and error feedback.

Fig. 5 is a chart depicting a method for determining a possible hierarchy of contexts (causal templates around a coded pixel marked by '?') in which optimal modeling context is selected.

30 Fig. 6 is a two-part graph illustrating a sign flipping technique to sharpen conditional error probability density functions for entropy reduction without increasing modeling space.

35 Fig. 7 is the block diagram of a four-pass encoding scheme for progressive transmission according to the invention.

Fig. 8 is a diagram and key illustrating interlaced sampling for the first pass and the configuration of prediction and modeling context for such a pass.

5 Fig. 9 is a diagram and key illustrating interlaced sampling for the second pass and the configuration of prediction and modeling context for such a pass.

10 Fig. 10 is a diagram and key illustrating functionally lossless reconstruction at the third pass and the configuration of prediction and modeling context used in this pass.

DESCRIPTION OF SPECIFIC EMBODIMENTS

Basic Compression System

15 Referring to Fig. 1, before actual compression, a system 10 operating according to the invention includes a decision maker 12 which dynamically selects on parts of the image between a compression scheme for continuous-tone image data and a compression scheme for binary image data based on context of the pixel data. In binary mode, the image data is
20 compressed by a first encoder 14 using context-based, adaptive, ternary entropy encoding. In the continuous-tone mode, the image data is compressed by a second encoder 16 using context-based, adaptive, non-linear predictive coding. Before encoding or decoding an image, the system is designed
25 to provide or selects between sequential transmission and progressive transmission, to encode or decode image data by the appropriated types of context-sensitive entropy-encoding and decoding processors. Image data encoded and decoded in raster scan order in a single pass through the image is
30 preferably transmitted by sequential transmission. The coding process uses prediction contexts that involve only the previous two or three scan lines of coded pixels. Consequently, the encoding and decoding processes require only a simple double buffer 18 that holds only the two or three
35 rows of pixels that immediately precede the current pixel. (Where greater buffer space is available, the system can achieve progressive transmission via interlaced subsampling

that encodes, and accordingly decodes, an image in four passes.)

In coding a picture, the CALIC system dynamically operates in two modes: binary and continuous-tone modes. The binary mode is for the situation in which the current locality of interest in the input image has no more than two distinct intensity values; hence it is designed for a more general class of images or subimages than the class of black-and-white images.

An innovation of the system is to distinguish between binary and continuous-tone types of pictures on a local, rather than a global, basis. The system automatically selects one of the two modes depending on the context of the current pixel. In the binary mode, the context-based adaptive entropy coder 14 is used to code three symbols, including an escape symbol. This distinction between binary and continuous-tone modes is important because the coding methodologies are very different in the two modes, and it contributes to the universality and robustness of the CALIC system over a wide range of pictures, including those that mix texts, graphics and photographs in multimedia applications.

The selection between the continuous-tone and binary modes is based on pixel context. The mode selection is automatic and completely transparent to the user. No side information about mode switching is required. The coded image only needs a header of five bytes: two for width, two for height, and one for depth or the number of bits in intensity resolution.

In the continuous-tone mode, a mechanism 20 is provided to compress pixel values by context-based, adaptive, predictive coding. To predict the current pixel value, a predictor 22 is provided for a local intensity gradient to be estimated based on the neighboring pixels in horizontal and vertical directions. These estimates are used to detect the magnitude and orientation of edges in the input image so that adjustments can be made to the prediction accordingly. This gradient-adjusted prediction (GAP), denoted by \hat{I} , of I is further corrected based on the particular pattern that the

neighboring pixels exhibit (as shown here as an output of an error modeling function 26). This is necessary because gradients alone cannot adequately characterize some of the more complex relationships between the predicted pixel and its surroundings. Context modeling can exploit higher-order structures such as texture patterns in the image. This is done by quantizing the neighborhood into different conditioning classes, and estimating the expectation of the GAP prediction error $e = I - \hat{I}$ in each of these classes separately. The reasons for estimating conditional expectations rather than the conditional density functions, as commonly suggested in the literature, are to prevent the problem of context dilution (sample counts are insufficient to obtain good estimates), and to reduce time and space complexities. The conditional error expectation estimate is then fed back to the GAP predictor 22 \hat{I} to generate a second and improved prediction denoted by \ddot{I} , which is equal to $\hat{I} + \bar{e}$. The net effect is a context-based, adaptive, non-linear predictor that can correct itself by learning from its mistakes made in the past and in the same context. This is a key feature to distinguish the prediction scheme of the CALIC system from the existing prediction schemes.

The final prediction error $\epsilon = I - \ddot{I}$ is entropy encoded. In driving the entropy coder 16, a modest number (between 8 and 32) of conditional probabilities of prediction errors ϵ are estimated (by an estimator 24) depending on an estimate of the prediction error energy. The error energy estimator is a linear function of the horizontal and vertical gradient and of the prediction errors at the preceding pixels. It can be optimized off line via linear regression for specific types of images. The conditioning classes for entropy coding are generated by quantizing the error energy estimator. The quantization can also be optimized via off-line dynamic programming to minimize the total code length of a set of training images. In addition, the CALIC system uses a novel and elegant technique to "sharpen" the conditional probabilities (thus reducing underlying entropy)

for entropy coding by a simple sign flipping mechanism 28 without any increase in modeling space.

Gradient-Adjusted Predictor

5 A nonlinear predictor according to the invention adapts itself to the image gradients near the predicted pixel. This predictor improves the robustness of traditional DPCM predictors, particularly in areas of edges. The basic function is now explained.

10 Denote an input image of width W and height H by $I[i,j]$, $0 \leq i < W$, $0 \leq j < H$. In order not to obscure the concept of the proposed compression algorithm, we only consider in the following development the sequential coding of interior pixels $I[i,j]$, $2 \leq i < W-2$, $2 \leq j < H-2$. Many possible treatments of boundary pixels are possible, and they
15 do not make a significant difference in the final compression ratio due to the small population of boundary pixels. For instance, boundary pixels can be coded by a simple DPCM scheme as they are encountered in the raster scan.

20 To facilitate the prediction of $I[i,j]$ and entropy coding of the prediction error via context modeling, we compute the following quantities:

$$d_h = |I[i-1,j] - I[i-2,j]| + |I[i,j-1] - I[i-1,j-1]| + |I[i+1,j-1] - I[i,j-1]| \quad (2)$$

$$d_v = |I[i-1,j] - I[i-1,j-1]| + |I[i,j-1] - I[i,j-2]| + |I[i+1,j-1] - I[i+1,j-2]|.$$

25

Clearly, d_h and d_v are estimates, within a scaling factor, of the gradients of the intensity function near pixel $I[i,j]$ in horizontal and vertical directions. The values of d_h and d_v are used to detect the magnitude and orientation of
30 edges in the input image, and make necessary adjustments in the prediction. We aim to alleviate the problem that the precision of existing DPCM-type predictors can be adversely

affected by edges. In Equations (2) three absolute differences are used for d in each direction. This has been found to give the best compression results. Two or one absolute differences can be used here for lower complexity
5 with a small loss in performance. Efficient incremental and/or parallel schemes for evaluating d_h and d_v are straightforward. For instance, to avoid unnecessary repeated computations, one can store the values of $|I[\cdot, \cdot] - I[\cdot, \cdot]|$ associated with preceding pixels for future reference. This
10 only requires an array of the size W .

For simple denotation in the sequel, we let

$$\begin{aligned} n &= I[i, j-1], w = I[i-1, j], ne = I[i+1, j-1], \\ nw &= I[i-1, j-1], nn = I[i, j-2], ww = I[i-2, j], \end{aligned} \quad (3)$$

15

meaning north, west, northeast, northwest, north-north, and west-west, respectively.

The locations of these pixels with respect to $I[i, j]$ are given in Fig. 2.

20 Based on d_h and d_v , a simple technique, as described by the following conditional statements, is used to make a gradient-adjusted prediction (GAP) $\hat{I}[i, j]$ of $I[i, j]$.

11

```

IF ( $d_v - d_h > 80$ ) {sharp horizontal edge}
     $\hat{I}[i,j] = w$ 
ELSE IF ( $d_v - d_h < -80$ ) {sharp vertical edge}
     $\hat{I}[i,j] = n$ 
5   ELSE {
         $\hat{I}[i,j] = (w+n)/2 + (ne-nw)/4;$ 
    IF ( $d_v - d_h > 32$ ) {horizontal edge}
         $\hat{I}[i,j] = (\hat{I}[i,j]+w)/2$ 
    ELSE IF ( $d_v - d_h > 8$ ) {weak horizontal edge}
10     $\hat{I}[i,j] = (3\hat{I}[i,j]+w)/4$ 
    ELSE IF ( $d_v - d_h < -32$ ) {vertical edge}
         $\hat{I}[i,j] = (\hat{I}[i,j]+n)/2$ 
    ELSE IF ( $d_v - d_h < -8$ ) {weak vertical edge}
15     $\hat{I}[i,j] = (3\hat{I}[i,j]+n)/4$ 
    }

```

The procedure given above is parallelizable. This technique differs from the existing linear predictors in that it weights the neighboring pixels of $I[i,j]$ according to the estimated gradients of the image. In effect, $\hat{I}[i,j]$ is a
 20 simple, adaptive, nonlinear predictor. The predictor coefficients and thresholds given above were empirically chosen. A major criterion in choosing these coefficients is the ease of computations. For instance, most coefficients are of power of 2 to avoid multiplication and divisions. The
 25 multiples of 3 in the procedure above can be computed by a bit shift and an addition.

It is possible, albeit quite expensive, to optimize the coefficients and thresholds for an image or a class of images, so that a norm of the expected prediction error
 30 $E\{\|I - \hat{I}\|\}$ is minimized. It is not recommended that such an optimization process to be carried out on an image-by-image basis. However, it is important to point out that the coefficients and thresholds in computing $\hat{I}[i,j]$ can be set by the user, if the user knows the optimal or nearly optimal
 35 coefficients and thresholds for the target images.

Error Energy Quantization for Minimum Entropy

Although the nonlinear predictor $\hat{I}[i,j]$ outperforms linear predictors, it does not completely remove the statistical redundancy in the image. The variance of prediction errors $e = I - \hat{I}$ strongly correlates to the smoothness of the image around the predicted pixel $I[i,j]$. To model this correlation at a small computational cost, we define an error energy estimator to be

$$\Delta = ad_h + bd_v + c|e_w|, \quad (4)$$

10

where d_h and d_v are defined in Equation (2) which are reused here to quantify the variability of pixel values, and $e_w = I[i-1,j] - \hat{I}[i-1,j]$. $|e_w|$ is chosen because large errors tend to occur consecutively. The coefficients a , b , and c can be determined, in an off-line design process, by the standard linear regression technique so that Δ is the least-squares estimator of the error strength $|e|$ based on d_h , d_v , $|e_w|$. For algorithm efficiency, we recommend $a = b = 1$ and $c = 2$ in Equation (4). In our experiments we also find that other definitions of Δ can sometimes estimate the error energy better, such as

20

$$\Delta = a_{\min} \{d_h, d_v\} + b_{\max} \times \{|e_w|, |e_n|\}, \quad (5)$$

25

depending on images, where $e_n = I[i,j-1] - \hat{I}[i,j-1]$.

By conditioning the error distribution on Δ , we can separate the prediction errors into classes of different variances. Thus entropy coding of errors using estimated conditional probability $p(e|\Delta)$ improves coding efficiency over using $p(e)$. For time and space efficiency, we quantize Δ to L levels. In practice, $L=8$ is found to be sufficient. Larger L improves coding efficiency only marginally. Denote the Δ quantizer by Q , i.e., $Q(\Delta)$ in $\{0,1,\dots,7\}$. In entropy coding of prediction errors, we estimate and use eight conditional probabilities $p(e|Q(\Delta))$. Since Δ is a random variable, it requires only scalar quantization. The

30

35

errors. In an off-line design process, we get a training set of (e, Δ) pairs from test images, and use the standard dynamic programming technique to choose $0 = q_0 < q_1 < \dots < q_{L-1} < q_L = \infty$ to partition Δ into L intervals such that

5

$$- \sum_{d=0}^{L-1} \sum_{q_d \leq \Delta \leq q_{d+1}} p(e) \log p(e) \quad (6)$$

is minimized.

In practice, we found that an image-independent Δ quantizer whose bins are fixed,

$$q_1=5, q_2=15, q_3=25, q_4=42, q_5=60, q_6=85, q_7=140. \quad (7)$$

10

worked almost as well as the optimal image-dependent Δ quantizer.

15

Estimating $L=8$ conditional error probabilities $p(e|Q(\Delta))$ requires only a modest amount of memory in the phase of entropy coding of prediction errors. Furthermore, the small number of involved conditional error probabilities means that even small images will provide enough samples for context modeling to learn $p(e|Q(\Delta))$ quickly in adaptive entropy coding.

20

Context Modeling of Prediction Errors

25

The precision of the nonlinear GAP predictor $\hat{I}[i, j]$ can be significantly improved via context modeling, because gradients alone cannot adequately characterize some of the more complex relationships between the predicted pixel $I[i, j]$ and its surrounding. Context modeling of the prediction error $e = I - \hat{I}$ can exploit higher-order structures such as texture patterns in the image for further compression gains.

30

Denote by a K -tuple $C = \{x_0, x_1, \dots, x_{K-1}\}$ a modeling context of $\hat{I}[i, j]$ that consists of K events x_i .

For both space and time efficiencies and to avoid the problem of context dilution, we need to drastically reduce

the number of possible contexts C by vector quantization of C . Specifically, we consider a prediction context of eight events

$$C = \{x_0, \dots, x_6, x_7\} = \{n, w, nw, ne, nn, ww, 2n-nn, 2w-ww\} \quad (8)$$

- 5 and quantize context $C = \{x_0, x_1, \dots, x_7\}$ to an 8-bit binary number $b = b_7 b_6 \dots b_0$ using the prediction value \hat{I} as the threshold, namely

$$b_k = \begin{cases} 0 & \text{if } x_k \geq \hat{I}[i, j] \\ 1 & \text{if } x_k < \hat{I}[i, j] \end{cases}, \quad 0 \leq k < K = 8. \quad (9)$$

- Clearly, number B (a binary vector codeword) encodes the texture patterns in the modeling context which are indicative of the behavior of e . Also note that an event x_i in a prediction context need not be a neighboring pixel to $I[i, j]$. It can be a function of some neighboring pixels. By letting $x_6 = 2n-nn$, $x_7 = 2w-ww$ in the above, and consequently setting b_6 and b_7 depending on whether $2n-nn < \hat{I}[i, j]$ and $2w-ww < \hat{I}[i, j]$, we can detect whether the prediction value $\hat{I}[i, j]$ forms a convex or concave waveform with respect to the neighboring pixels in vertical and horizontal directions, as depicted by Fig. 3. This convexity information is useful in the context modeling of e .

- Since the variability of neighboring pixels also influences the error distribution, we combine the quantized error energy $0 \leq \lfloor Q(\Delta)/2 \rfloor < L/2$ with the quantized texture pattern $0 \leq B < 2^K$ to form compound prediction contexts, denoted by $C(\lfloor Q(\Delta)/2 \rfloor, B)$. This scheme can be viewed as a product quantization of two independently treated image features: spatial texture patterns and the energy of prediction errors.

- At a glance, we would seemingly use $4 \cdot 2^8 = 1024$ different compound contexts, since $0 \leq \lfloor Q(\Delta)/2 \rfloor < L/2 = 4$ and $0 \leq B < 2^K = 2^8$. However, not all 2^8 binary codewords of B

quantizer defined by (9) are possible. If the prediction value $\hat{I}[i,j]$ lies in between n and nn , then the condition $2n - nn < \hat{I}[i,j]$ is either always true or always false. In other words, the outcome of the test $2n - nn < \hat{I}[i,j]$ in (9) is uniquely determined if $n > \hat{I}[i,j]$ and $nn < \hat{I}[i,j]$, or if $n < \hat{I}[i,j]$ and $nn > \hat{I}[i,j]$. This should be apparent from Fig. 3. By referring to (8) and (9), we see that in B , b_6 is fixed if b_2 and b_5 are different. Likewise, b_7 is fixed if b_0 and b_4 are different. Because b_6 , b_2 , and b_5 are not independent, they can only yield $2 \cdot 2 + 2 \cdot 1 = 6$ combinations. The same holds for b_7 , b_0 , and b_4 . Thus, B represents only $6 \cdot 6 \cdot 4 = 144$ rather than 256 valid texture patterns. Finally, the total number of valid compound contexts in error modeling is only $4 \cdot 144 = 576$.

The model cost is still too high to estimate $p(e|C(\lfloor Q(\Delta)/2 \rfloor, B))$ for 576 compound contexts. Therefore, we estimate only the conditional expectations $E\{e|C(\delta, \beta)\}$ using the corresponding sample means $\bar{e}(\delta, \beta)$ for different compound contexts. Computing $\bar{e}(\delta, \beta)$ involves only accumulating the error terms in each compound context and maintaining a count on the occurrence of each context. For each compound context $C(\delta, \beta)$, $0 \leq \delta < 4$, $0 \leq \beta < 144$, we count the number $N(\delta, \beta)$ of occurrences of $C(\delta, \beta)$ for the purpose of computing the conditional sample mean $\bar{e}(\delta, \beta)$. We only use one byte to store the occurrence count. Whenever the count reaches some value $n_{\max} < 256$, we scale down the count but let the corresponding $\bar{e}(\delta, \beta)$ intact. We recommend $n_{\max} = 128$.

In order to update $\bar{e}(\delta, \beta)$ in error modeling, we use an accumulator $S(\delta, \beta)$ of $e(\delta, \beta)$, and compute

$$\bar{e}(\delta, \beta) = S(\delta, \beta) / N(\delta, \beta) \quad (10)$$

whenever a compound context $C(\delta, \beta)$ occurs. We use a 16-bit signed integer to store $S(\delta, \beta)$ whose range is more than sufficient in practice. It takes more than 128 consecutive $\bar{e}(\delta, \beta) = -255$ or $\bar{e}(\delta, \beta) = 255$ to violate the bounds

$-2^{15} < S(\delta, \beta) < 2^{15}$, but we have $n_{\max} = 128$. Once $N(\delta, \beta)$ reaches 128, we re-scale the variables by setting

$$S(\delta, \beta) = S(\delta, \beta) / 2; \quad N(\delta, \beta) = 64 \quad (11)$$

In summary, the coding system needs one byte for $N(\delta, \beta)$ and two bytes for $S(\delta, \beta)$ for each of 576 compound contexts. Thus the total size of the memory required by context modeling of prediction errors can be as small as 1728 bytes.

Besides being a technique to reduce the memory use, rescaling also has the beneficial side effect of aging the observed data. This is an inexpensive way of adapting the context error modeling to time varying sources. Indeed, the scaling technique gave a slightly better compression in our experiments.

Context-based Adaptive Predictor via Error Feedback

The idea of gaining coding efficiency by context modeling of expected errors $E\{e|C(\delta, \beta)\}$ came from the observation that the conditional mean $\bar{e}(\delta, \beta)$ is generally not zero in a given context $C(\delta, \beta)$. This does not contradict the well-known fact that the DPCM prediction error without conditioning on the contexts, follows a Laplacian (symmetric exponential) distribution, hence is zero mean for most continuous-tone images. The Laplacian distribution can be viewed as a composition of distributions of different means and different variances. Context modeling of prediction errors is a means to separate these distributions. Therefore, the more biased is $\bar{e}(\delta, \beta)$ from zero, the more effective is the modeling. Since the conditional mean $\bar{e}(\delta, \beta)$ is the most likely prediction error in a given compound context $C(\delta, \beta)$, we can improve the prediction by feeding back $\bar{e}(\delta, \beta)$ and adjusting the prediction \hat{I} to $\tilde{I} = \hat{I} + \bar{e}(\delta, \beta)$.

In order not to over-adjust the GAP predictor, we actually consider the new prediction error $\epsilon = \hat{I} - \tilde{I}$ rather than $e = I - \hat{I}$ in context-based error modeling. The context modeling of ϵ in turn leads to an improved predictor for

$I: I = \hat{I} + \bar{\epsilon}(\delta, \beta)$ where $\bar{\epsilon}(\delta, \beta)$ is the sample mean of ϵ conditioned on compound context $C(\delta, \beta)$.

Conceptually, we have a two-stage adaptive prediction scheme via context modeling of errors and error feedback. A block diagram of the scheme is given in Fig. 4. In the first stage, the encoder makes a GAP prediction \hat{I} based on d_h and d_v . Then \hat{I} is used as a threshold value to quantize a block of neighboring pixels into a spatial texture pattern B as in (9). The encoder also computes the error energy estimator Δ , and quantizes it to $Q(\Delta)$. By a simple Cartesian product of $Q(\Delta)$ and B , a compound context $C(\lfloor Q(\Delta)/2 \rfloor, B) = C(\delta, \beta)$ is formed for context modeling of prediction errors. Algorithmically, $C(\delta, \beta)$ just serves as an index to look up the most likely prediction error $\bar{\epsilon}(\delta, \beta)$ from a table of conditional sample means. In the second stage, $\bar{\epsilon}(\delta, \beta)$ is added to the first prediction \hat{I} to arrive at an adaptive, context-based, non-linear prediction for $I[i, j]$:

$$\hat{I}[i, j] = \hat{I}[i, j] + \bar{\epsilon}(\delta, \beta). \quad (12)$$

Finally, the error $\epsilon = I[i, j] - \hat{I}[i, j]$ of the second prediction is entropy coded. Upon receiving ϵ , the decoder can recover $I[i, j]$ via the same adaptive context modeling scheme. After each pixel is encoded/decoded, both the encoder and decoder update $\bar{\epsilon}(\delta, \beta)$ by the most recent observation $\epsilon = I - \hat{I}$ in the current compound context $C(\delta, \beta)$.

Context Hierarchy and Context Selection

We call the context C defined by (8) the kernel context. It captures the most of the correlation between the prediction error and the surrounding pixels. However, the kernel context is generally not optimal in the sense of obtaining the minimum description length. The technique introduced in this section is for selecting optimal model contexts in a context hierarchy for gaining maximum compression.

In his algorithm called Context Rissanen proposed a method of choosing optimal modeling contexts by assuming that

a one-dimensional data stream like text is generated by a tree model. However, selecting optimal modeling contexts in universal lossless image compression is much more difficult than in universal text compression, because the image signal is not one-dimensional. In order to use the sequential tree model on two-dimensional image data, one has to impose an order on the pixels in a modeling template. This order is required when traversing the tree for selecting the "optimal" context in which the next symbol is coded. However, as a modeling state, a context (causal template) is an integral entity, and should be treated as such. Any ordering of pixels in the causal template is artificial, and the modeling context selected based on this ordering may compromise the coding efficiency.

To retain the cohesion and integrity of pixels in a modeling context, we generalize the nodes of the tree model to a two-dimensional causal template, and call the resulting tree model context hierarchy as depicted by Fig. 5. The root of the context hierarchy is the kernel context, denoted by C_0 .

Given a modeling context C_0 that is a set of pixels, we call context C_1 an expansion of C_0 if $C_1 \supset C_0$. The set containment relation naturally defines a context hierarchy. The nodes of the context hierarchy correspond to nested modeling contexts.

Although the conditional entropy with respect to the error probability density function $p(e|C)$ is monotonically non-increasing in the size of causal template C , larger modeling context does not necessarily give more accurate estimate of e_i . Too large a causal template spreads samples too thin into too many contexts to obtain reliable estimates. Intuitively speaking, we want to choose a context as large as possible but provided that it occurred for a sufficient number of times in the past. In principle, Rissanen's algorithm Context can be generalized to the context hierarchy for selecting optimal context. Unfortunately, algorithm Context is impractical, and not suitable for hardware implementation or parallelization. As a practical alternative, we invented the following simple context selection method.

Refer to Fig. 5, we have $C_0 \subset C_1 \subset C_3$, and $C_0 \subset C_2 \subset C_4$, where $C_0 = \{n, w, nw, ne, nn, ww, 2n-nn, 2w-ww\}$ is the kernel context. For each context C_i , $0 \leq i \leq 4$, maintain the count N_i of the number of the times that C_i has happened, and
 5 keep the sum S_i of the prediction errors under C_i , as described in section 4.3. Then the optimal context selection and the corresponding context-based, adaptive, non-linear prediction can be efficiently done by the following simple logic:

10

$$S_1 = S_r = S_0;$$

$$N_1 = N_r = N_0;$$

$$\{\text{IF}\} (N_3 > \tau)$$

$$\{S_1 = S_3; N_1 = N_3;\}$$

15

$$\{\text{ELSE IF}\} (N_1 > \tau)$$

$$\{S_1 = S_1; N_1 = N_1;\}$$

$$\{\text{IF}\} (N_4 > \tau)$$

$$\{S_r = S_4; N_r = N_4;\}$$

$$\{\text{ELSE IF}\} (N_2 > \tau)$$

20

$$\{S_r = S_2; N_r = N_2;\}$$

$$\hat{f} = \hat{f} + \frac{S_1 + S_r}{N_1 + N_2};$$

25

The threshold τ is normally set to 7. The value of τ should be proportional to the variance of prediction errors.

Sharpening of Conditional Error Probabilities

30

As described in Section 4.2, we use the error energy quantizer to form eight conditional error probabilities $p(\epsilon|Q(\Delta))$, $0 \leq Q(\Delta) < 8$, for entropy coding of ϵ . Recall that the error energy quantizer Q is optimized in (6) to minimize the combined entropy of the eight conditional error
 35 probabilities. But $p(\epsilon|Q(\Delta))$ as is does not benefit from the error modeling based on the texture pattern B . For operational reasons, $p(\epsilon|Q(\Delta))$ rather than $p(\epsilon|C(\lfloor Q(\Delta)/2 \rfloor, B))$

are used in entropy coding of ϵ , as the number of different compound contexts is too large.

However, on a second reflection, the conditional sample means $\bar{\epsilon}(\delta, \beta)$, which are dynamically computed for error feedback, can "sharpen" the conditional probabilities
 5 $p(\epsilon|Q(\Delta))$, or reduce their variances, as explained below, for free! We observed that under two different contexts $C(\delta, \beta_1)$ and $C(\delta, \beta_2)$, two conditional sample means $\bar{\epsilon}(\delta, \beta_1)$ and $\bar{\epsilon}(\delta, \beta_2)$ could be significantly off from zero and have opposite signs,
 10 reflecting very different $p(\epsilon|C(\delta, \beta_1))$ and $p(\epsilon|C(\delta, \beta_2))$. For a fixed $0 \leq \delta < 8$, one can then split the conditional probability $p(\epsilon|Q(\delta) = \delta)$ into two

$$p_+(\epsilon|\delta) = \frac{\sum_{\beta} p(\epsilon|\bar{\epsilon}(\delta, \beta) \geq 0)}{\sum_{\epsilon} \sum_{\beta} (\epsilon|\bar{\epsilon}(\delta, \beta) \geq 0)} \quad (13)$$

$$p_-(\epsilon|\delta) = \frac{\sum_{\beta} p(\epsilon|\bar{\epsilon}(\delta, \beta) < 0)}{\sum_{\epsilon} \sum_{\beta} (\epsilon|\bar{\epsilon}(\delta, \beta) < 0)}$$

15 p_+ being the conditional probability of the errors ϵ in contexts $C(\delta, \beta)$ such that $\bar{\epsilon}(\delta, \beta) \geq 0$, and p_- being such that $\bar{\epsilon}(\delta, \beta) < 0$. The two resulting conditional probabilities p_+ and p_- will have smaller variances than $p(\epsilon|Q(\delta) = \delta)$, making more distinctive error statistics available to the entropy
 20 coder. Seemingly, this splitting of conditional probabilities for entropy coding would double the memory use. But a very simple technique can reap almost the same compression gain achievable by (13) without increasing the use of physical memory.

25 The conditional probability $p(\epsilon|\delta)$ is the mixture of $p_+(\epsilon|\delta)$ and $p_-(\epsilon|\delta)$. We observed in our experiments that $p_+(\epsilon|\delta)$ and $p_-(\epsilon|\delta)$ are approximately mirror image of each other around the origin. Therefore, we can flip $p_-(\epsilon|\delta)$ around the origin and superimpose it onto $p_+(\epsilon|\delta)$ to create a
 30 more biased conditional probability for entropy coding:

$$\hat{p}(\epsilon|\delta) = p_+(\epsilon|\delta) + p_-(-\epsilon|\delta) \quad (14)$$

as illustrated in Fig. 6. Clearly, $\hat{p}(\epsilon|\delta)$ has a lower entropy than $p(\epsilon|\delta)$, and it can be used by an entropy coder to reduce bit rates without any extra space. Suppose that the current
5 context is $C(\delta, \beta)$. Before encoding $\epsilon = I - \tilde{I}$, the encoder checks whether $\bar{\epsilon}(\delta, \beta) < 0$. If yes, $-\epsilon$, otherwise, ϵ is encoded. Since the decoder also knows $C(\delta, \beta)$ and $\bar{\epsilon}(\delta, \beta)$, it can reverse the sign, if necessary, to reconstruct ϵ .

This simple technique of sign flipping can reduce
10 the bit rate by as much as three per cent on some test images. The improved coding efficiency is obtained, because the error modeling based on compound contexts $C(\delta, \beta)$ captures some of the statistical redundancy in the sign of ϵ . In essence, by preserving or reversing the sign of ϵ , we make a prediction on
15 the sign of ϵ . Thanks to context modeling the prediction is correct more than half of the time, yielding compression gains.

Binary Mode

20 The above adaptive, predictive coding scheme via context modeling of prediction errors is designed for compressing natural continuous-tone images. But the coder performance can be quite poor on images or subimages which contain very few grey levels, especially when compared with an
25 algorithm based on explicit Markov modeling. The latter becomes feasible precisely because of the very small size of the symbol set. In this situation it is more effective to code pixel values directly rather than prediction errors.

In order to effectively compress uniform or nearly
30 uniform image areas, graphics, rasterized documents, and any combination of natural images with one or more of these types, according to the invention there is a binary coder as a separate module of the compression system, and means are provided to select between modes. The system operates in two
35 modes: continuous-tone and binary. Before $I[i, j]$ is to be coded, a tester is invoked to for proper mode. It checks six

neighboring pixels: ww, w, nw, n, ne, nn . If these six pixels have no more than two different values, the binary mode is triggered; otherwise the system is caused to enter the continuous-tone mode in which the predictor \tilde{I} and the compound contexts $C(\delta, \beta)$ are in use.

In the binary mode, let $s_1 = w$ and let the other value, if any, be s_2 . The encoder describes $I[i, j]$ in one of the three states, using a ternary code T :

$$T = \begin{cases} 0 & \text{if } I[i, j] = s_1 \\ 1 & \text{if } I[i, j] = s_2 \\ 2 & \text{otherwise} \end{cases} \quad (15)$$

In the escape case of $T=2$, the encoder switches from the binary mode to the continuous-tone mode. This happens when the actual $I[i, j]$ is found to violate the condition for binary mode.

Similar to the continuous-tone mode, upon entering the binary mode, the system considers a context of six events

$$C = \{x_0, x_1, \dots, x_5\} = \{w, n, nw, ne, ww, nn\}, \quad (16)$$

and quantize context $C = \{x_0, x_1, \dots, x_5\}$ to a 6-bit binary number $B = b_5 b_4 b_3 b_2 b_1 b_0$:

$$b_k = \begin{cases} 0 & \text{if } x_k = s_1 \\ 1 & \text{if } x_k = s_2 \end{cases}, \quad 0 \leq k < 6. \quad (17)$$

The binary number B represents a texture pattern around $I[i, j]$. Note that because $s_1 = w$, $b_0 = 0$ can be ignored from the texture encoding B . Thus there are only $2^5 = 32$ different prediction contexts in the binary mode. Consequently, we use 32 conditional probabilities $p(T|B)$ to drive an adaptive arithmetic coder of three symbols. For the context modeling in the binary mode, only $32 \times 3 = 96$ frequency counts need to be maintained.

The incorporation of the binary mode into the compression system adds little to the system complexity, while greatly enhancing the coding performance on binary-type images, multimedia images, and images with large uniform areas. An important feature of our design is that the switch between continuous-tone and binary modes is context-dependent, automatic, and completely transparent to the user.

Variable Image Resolutions

All parameters and thresholds of the predictors and the context quantizers are designed and optimized for images of 8 bit resolution, because this is the most popular intensity resolution. However, we devised a very simple scaling mechanism for the coding system to handle images of all resolutions with ease, with negligible penalty on compression performance.

It is a design decision to scale d_h and d_v by some factor λ , if necessary, in order to make the prediction module and context modeling module invariant to intensity resolutions. This is because it is far more involved to adjust parameters and thresholds in the prediction and modeling modules for the best compression performance for different intensity resolutions.

Suppose that the input image has an intensity resolution of z bits, $1 < z \leq 16$. The simplest scaling factor for d_h and d_v is $\lambda = 2^{8-z}$. But in our experiments, we found that this simple scaling factor can incur quite heavy loss in coding efficiency if the input images are very smooth relatively to the dynamic intensity range.

To solve this problem, it is preferred that d_h and d_v be scaled according to the standard deviation σ of the errors (the average of error magnitudes) in the previous row. Computationally, the value of σ for the previous row is easy to update during the raster scan, requiring only an accumulator. Based on σ , we use an empirical formula

$$\lambda = 2^{-\lfloor \frac{s-8}{2} \rfloor - \max(0, \lfloor \log_2 s-5 \rfloor)} \quad (18)$$

to compute the scaling factor for d_h and d_v .
Note that the scaling requires only binary shifting.

5 Progressive Transmission

To achieve progressive transmission, the encoding, and accordingly the decoding, of an image is done in four passes as schematically described by Fig. 7. Each successive pass presents a better approximation of the original at a higher bit rate. The first pass obtains a compression ratio from 9:1 to 7:1 depending on images with an image quality that is comparable with lossy JPEG at the same compression ratio. The second pass presents a compressed image that is visually indistinguishable from the original at a compression ratio around 5:1. The third pass yields so-called functionally lossless compression at a compression ratio around 3:1. After the third pass, the errors are below such tight bounds in both magnitude and spatial distribution that their effects on most image processing and analysis functions are provably negligible. The fourth pass reaches true lossless compression by adding no more than 0.25 bits per pixel to the functionally lossless image from the third pass. The feature of progressive transmission is important for many applications such as remote query and retrieval in image databases, and in multiuse environment as described by US Patent 4,969,204 where different imaging devices of different resolutions access to the same image.

Unlike sequential encoding schemes, the multipass encoding of an image can generate completely enclosing prediction contexts.

Each of the four passes uses an interlaced sampling of the original image. Fig. 8 to Fig. 10 illustrate the patterns of the interlaced sampling from the first pass to the fourth pass respectively. Besides facilitating progressive transmission, the four-pass interlaced sampling scheme is also designed for the formation of modeling contexts that enclose

the predicted pixels in all directions. The spatial configurations of the resulting modeling contexts in individual passes are also depicted in Fig. 8 through Fig. 10. Many image features such as intensity gradients, edge orientations, textures can be better modeled by enclosing contexts as those in Fig. 9 and Fig. 10 than by the contexts that only partially enclose the unknown pixel as in one-pass sequential coding.

The first pass encodes a subsampled $W/2 \times H/2$ image denoted by I_1 , with the following relation between I and I_1

$$I_1[i, j] = \left\lfloor \frac{I[2i, 2j] + I[2i+1, 2j+1]}{2} \right\rfloor, \quad 0 \leq i < W/2, \quad 0 \leq j < H/2. \quad (19)$$

Note that $I_1[i, j]$ is not the average intensity of a 2×2 block but the average intensity of two diagonally-adjacent pixels. This subsampling is designed to benefit the context-based adaptive prediction in the subsequent passes. The $W/2 \times H/2$ image I_1 is encoded in the conventional raster scan sequence by the sequential CALIC algorithm as described in previous sections.

The second pass uses I_1 as the contexts to encode an $W/2 \times H/2$ array of 2-pixel blocks, called image I_2 , where

$$I_2[i, j] = (I[2i, 2j], I[2i+1, 2j+1]), \quad 0 \leq i < W/2, \quad 0 \leq j < H/2. \quad (20)$$

Note that $I_2[i, j]$ is a two-dimensional vector. Consequently, upon the completion of encoding I_2 , half of original $W \times H$ pixels will be reconstructed. Again, the image I_2 is encoded in the conventional raster scan sequence. Since I_1 is by now available to both the encoder and decoder, it provides completely surrounding contexts for predicting and modeling $I[2i, 2j]$. A possible context in this case is

$$C = \{I_1[i, j], I_1[i+1, j], I_1[i, j+1], I[2i-1, 2j+1], I[2i-1, 2j-1], \\ I[2i+1, 2j-1], I[2i-2, 2j], I[2i, 2j-2]\} \quad (21)$$

See Fig. 9 for the spatial configuration of the context.

Using the modeling contexts, the CALIC algorithm encodes the subsampled image I_2 .

Once $I[2i, 2j]$ is reconstructed, the decoder can set

$$I[2i+1, 2j+1] = 2I_1[i, j] - I[2i, 2j] \quad (22)$$

without receiving any information on $I[2i+1, 2j+1]$. But a consultation with (19) reveals that (22) can cause a rounding error of 1 if the sum $I[i, j] + I[2i+1, 2j+1]$ is an odd integer. The probability for this rounding error is 0.5. For the time being, this tiny, isolated error is tolerated. In Figs. 9 and 10, the pixel with a probability 0.5 of being 1 smaller than the original value is marked by the unfilled circle.

The third pass encodes the remaining half of the original $W \times H$ image I ; namely, pixels interlaced in the checker board pattern, $I[2i+1, 2j]$ and $I[2i, 2j+1]$, $0 \leq i < W/2$, $0 \leq j < H/2$. In the third pass, the image is also encoded in the conventional raster scan order. After the intermediate images I_1 and I_2 are coded, the modeling contexts available to the third pass are the most complete and relevant to the predicted pixels. Hence much more compression of the remaining half of the image can be achieved. Fig. 10 illustrates the situation in the third pass and depicts a configuration of modeling context among many possibilities.

After the third pass, the image is said to be functionally lossless for the following reasons:

- The error magnitude is bounded by 1.

- The errors are spatially isolated. The errors can only occur at the positions marked by unfilled circles in Fig. 4. No consecutive errors can happen.
- At the already isolated potential error positions, the errors are distributed randomly.
- The errors have no effect on the average intensity in a window of any size.
- The errors cannot alter the sign of the gradients of the original image function at any position or in any direction.
- The effects of the errors on the filtering operations on the original image will be either nil or negligibly small. Given a filter kernel, the worst error impact on the filtering operation can be precisely computed to be a known prior.

If true lossless reconstruction is indeed required, the rounding errors of (22) can be corrected by one more pass.

Progressive, multiresolution decompression is a natural product of the above four-pass encoding scheme. By choosing I_1 , I_2 , and then functionally lossless I or true lossless I , the decoder obtains a reconstruction at a desired resolution and bit rate.

Entropy Coding of Prediction Errors

An advantage of this invention is the clean separation between context-based prediction and modeling of image data and entropy coding of prediction errors. Any entropy coder, be it Huffman or arithmetic, static or adaptive, binary or m -ary, can be easily interfaced with the CALIC system. The system should be made flexible enough to accommodate all entropy coding methods that are known to date. If system complexity and cost are of primary concern, static global Huffman coders can be used. On the other hand, for

best compression possible, adaptive arithmetic coders will be the choice.

Decompression

5 The decompression or decoding process of the CALIC system is a straightforward reverse of the encoding process whose details were just explained. The decoding process reconstructs the original image in the same order as the image was encoded by the CALIC system. The decoding system selects
10 either sequential or progressive decompression, depending on whether the image is compressed sequentially or progressively. Referring to Fig. 1, the CALIC decoding process automatically enters one of the binary or continuous-tone decompression modes, based on the exactly same context-sensitive decision
15 that is made by the encoder. Thus the encoder and decoder can synchronize with each other without using any side information.

 In the binary mode, a context-based adaptive entropy coder that is identical to the one used in the encoding
20 process is used to decode the current symbol.

 In the continuous-tone mode, the gradient-adjusted prediction, denoted \hat{I} , of I is made exactly in the same manner as the encoder, and the predictor \hat{I} is further improved by the same context-sensitive error feedback technique that is used
25 in the encoding process. Since predictions for the current pixel that are made by the encoder are based solely on the pixels that have already been encoded, the decoder can perform the same predictions, context formation, context selection, context quantization, error feedback and entropy coding
30 context determination. The decoder can then uniquely decode the prediction error from the compressed bit stream and add it to the prediction in order to reconstruct the exact original value of the current pixel. The decoder proceeds in this manner and recovers the original image pixel by pixel from the
35 prediction errors that it decodes one at a time from the compressed bit stream.

 The invention has now been explained with reference to specific embodiments. Other embodiments will be apparent

to those of ordinary skill in the art. Therefore, it is not intended that this invention be limited, except as indicated by the appended claims.

WHAT IS CLAIMED IS:

1 1. A method for lossless image data compression
2 comprising the steps of:

3 dynamically and automatically choosing between a
4 binary level compression mode and a continuous-tone level
5 compression mode for each current image element based on local
6 context of said current image element; thereafter

7 compressing binary image data of said current image
8 element, in said binary mode, by context-based, adaptive,
9 ternary entropy encoding and storing first results in a memory
10 as a first compressed portion of the image data;

11 compressing continuous-tone level image data of said
12 current image element, in said continuous-tone mode, by
13 context-based, adaptive, non-linear predictive entropy
14 encoding and storing second results in said memory as a second
15 compressed portion of the image data;

16 wherein said binary mode entropy encoding is a first
17 selected context-sensitive entropy-encoding process; and

18 wherein said continuous-tone mode entropy encoding
19 is a second selected context-sensitive entropy-encoding
20 process wherein prediction error is entropy encoded.

1 2. The method according to claim 1 wherein first
2 selected specific compression techniques are used for
3 sequential transmission of said first compressed portion and
4 said second compressed portion of the image data.

5
6 3. The method according to claim 1 wherein second
7 selected specific compression techniques are used for
8 progressive transmission of said first compressed portion and
9 said second compressed portion of the image data.

1 4. The method according to claim 3 further
2 comprising subsampling images in four passes; and
3 forming from subsampling data enclosing contexts in
4 the form of causal templates.

1 5. The method according to claim 1 wherein said
2 current image element is a pixel, and wherein in said
3 continuous- tone mode compressing step comprises predicting
4 values for each said pixel based on local intensity gradient
5 estimates.

1 6. The method according to claim 5 wherein said
2 predicting step further includes estimating energy of
3 prediction errors based on both local gradient and prediction
4 errors of preceding pixels together to obtain an error energy
5 estimator.

1 7. The method according to claim 5 wherein said
2 continuous-tone mode entropy encoding further includes
3 estimating energy of prediction errors based on both local
4 gradient and prediction errors of preceding pixels together.

1 8. The method according to claim 7 further
2 including reversing sign of a prediction error to sharpen the
3 conditional error probability densities for entropy coding.

1 9. The method according to claim 7 further
2 including adaptive bit shifting in each encoding context in
3 order to scale d_h and d_v , where d_h and d_v are estimates,
4 within a scaling factor, of gradients of the intensity
5 function near pixel $I[i,j]$ in horizontal and vertical
6 directions, such that encoding schemes are independent of
7 intensity resolutions.

1 10. The method according to claim 6 further
2 including quantizing the error energy estimator to form
3 conditioning classes for entropy encoding in the continuous-
4 tone mode.

1 11. The method according to claim 5, further
2 including forming modeling contexts by causal templates using
3 a prediction value of a pixel.

1 12. The method according to claim 11, wherein said
2 quantizing of the modeling contexts comprises binarization of
3 spatial texture patterns.

1 13. The method according to claim 5, further
2 including selecting an optimal modeling context in a context
3 hierarchy.

1 14. The method according to claim 5 further
2 including combining a quantized texture pattern and a
3 quantized error energy estimator into a compound context for
4 modeling prediction errors.

1 15. The method according to claim 5 further
2 including using conditional expectations of the prediction
3 errors to model errors at low model cost and low computational
4 cost.

1 16. The method according to claim 5 further
2 including reducing weight of past statistics in adapting to
3 non-stationary image data.

1 17. The method according to claim 5 further
2 including correcting prediction via context modeling of
3 prediction errors and error feedback.

1 18. The method according to claim 1 wherein said
2 choosing step comprises selecting between said binary level
3 compression mode and said continuous-tone level compression
4 mode based on local context information only and without use
5 of side information.

1 19. The method according to claim 1 wherein said
2 binary mode compressing step comprises forming a modeling
3 context for adaptive binary entropy encoding.

1 20. A method for lossless image data compression
2 comprising the steps of:

3 dynamically and automatically choosing between a
4 binary level compression mode and a continuous-tone level
5 compression mode for each current image element based on local
6 context of said current image element, said current image
7 element being a pixel; thereafter,

8 compressing binary image data of said current image
9 element, in said binary mode, by context-based, adaptive,
10 ternary entropy encoding and storing first results in a memory
11 as a first compressed portion of the image data;

12 compressing continuous-tone level image data of said
13 current image element, in said continuous-tone mode, by
14 context-based, adaptive, non-linear predictive entropy
15 encoding and storing second results in said memory as a second
16 compressed portion of the image data;

17 wherein said binary mode entropy encoding is a first
18 selected context-sensitive entropy-encoding process; and

19 wherein said continuous-tone mode entropy encoding
20 is a second selected context-sensitive entropy-encoding
21 process wherein prediction error is entropy encoded;

22 wherein said continuous-tone mode compressing step
23 further includes:

24 predicting values for each said pixel based on local
25 intensity gradient estimates, including estimating energy
26 of prediction errors based on both local gradient and
27 prediction errors of preceding pixels together to obtain
28 an error energy estimator;

29 forming modeling contexts by causal templates using
30 a prediction value of a pixel;

31 selecting an optimal modeling context in a context
32 hierarchy;

33 combining a quantized texture pattern and a
34 quantized error energy estimator into a compound context
35 for modeling prediction errors;

36 using conditional expectations of the prediction
37 errors to model errors at low model cost and
38 computational cost;

39 correcting prediction via context modeling of
40 prediction errors and error feedback; and

41 wherein said continuous-tone mode entropy encoding
42 further includes:
43 estimating energy of prediction errors based on both
44 local gradient and prediction errors of preceding pixels
45 together;
46 quantizing the error energy estimator to form
47 conditioning classes for entropy encoding in the
48 continuous-tone mode; and
49 reversing sign of a prediction error to sharpen the
50 conditional error probability densities for entropy
51 encoding.

1 21. A system for lossless image data compression
2 comprising:
3 a first subsystem for dynamically and automatically
4 choosing between a binary level compression mode and a
5 continuous-tone level compression mode for each current image
6 element of an input image based on local context of said
7 current image element, including a values predictor operative
8 on each said image element wherein prediction is based on
9 local intensity gradient estimates;
10 a second subsystem for compressing, in said binary
11 mode, the image data of said current image element by
12 context-based, adaptive, ternary entropy encoding as a first
13 compressed portion of the image data; and
14 a third subsystem for compressing, in said
15 continuous-tone mode, the image data of said current image
16 element by context-based, adaptive, non-linear predictive
17 entropy encoding as entropy-encoded prediction errors of a
18 second portion of the image.

1 22. A method for lossless image data decompression
2 comprising the steps of:
3 dynamically and automatically choosing between image
4 data compressed by a binary level compression and image data
5 compressed by continuous-tone level compression for each
6 current image element based on local context of said current
7 image element; thereafter either

8 decompressing, in said binary mode, the image data
9 of said current image element by context-based, adaptive,
10 ternary entropy decoding to produce a first decompressed
11 portion of the image data; and

12 decompressing, in said continuous-tone mode, the
13 image data of said current image element by context-based,
14 adaptive, non-linear predictive entropy decoding as a second
15 decompressed portion of the image; and

16 directing the first decompressed portion and the
17 second decompressed portion to an output utilization device to
18 produce a binary level and continuous-tone level composite
19 image.

1 23. A method for functionally lossless image data
2 compression comprising the steps of:

3 dynamically and automatically choosing between a
4 binary level compression mode and a continuous-tone level
5 compression mode for each current image element based on local
6 context of said current image element; thereafter

7 compressing binary image data of said current image
8 element, in said binary mode, by functionally-lossless
9 context-based, adaptive, ternary entropy encoding and storing
10 first results in a memory as a first compressed portion of the
11 image data, such that errors in a decompressed version of said
12 image data do not exceed 1 and never occur consecutively in
13 image space;

14 compressing continuous-tone level image data of said
15 current image element, in said continuous-tone mode, by
16 functionally-lossless context-based, adaptive, non-linear
17 predictive entropy encoding and storing second results in said
18 memory as a second compressed portion of the image data, such
19 that errors in a decompressed version of said image data do
20 not exceed 1 and never occur consecutively in image space;

21 wherein said binary mode entropy encoding is a first
22 selected context-sensitive entropy-encoding process; and

23 wherein said continuous-tone mode entropy encoding
24 is a second selected context-sensitive entropy-encoding
25 process wherein prediction error is entropy encoded.

1/5

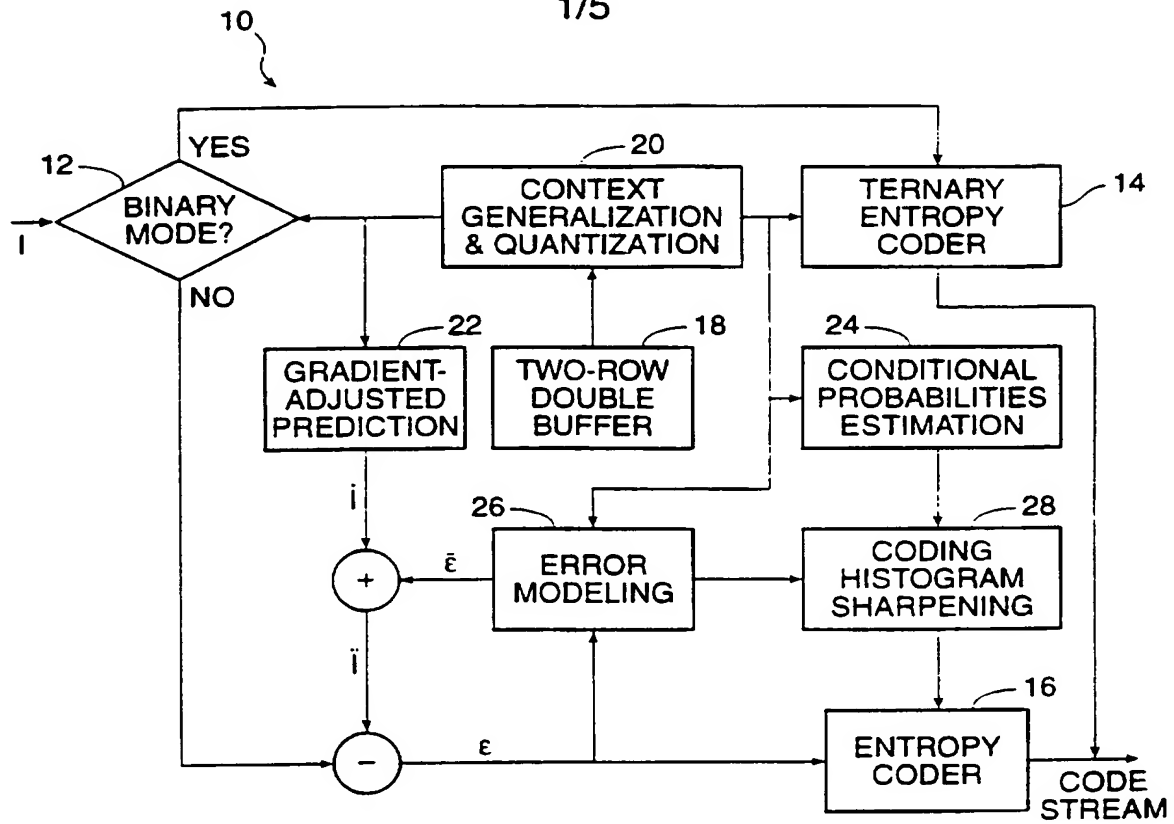


FIG. 1

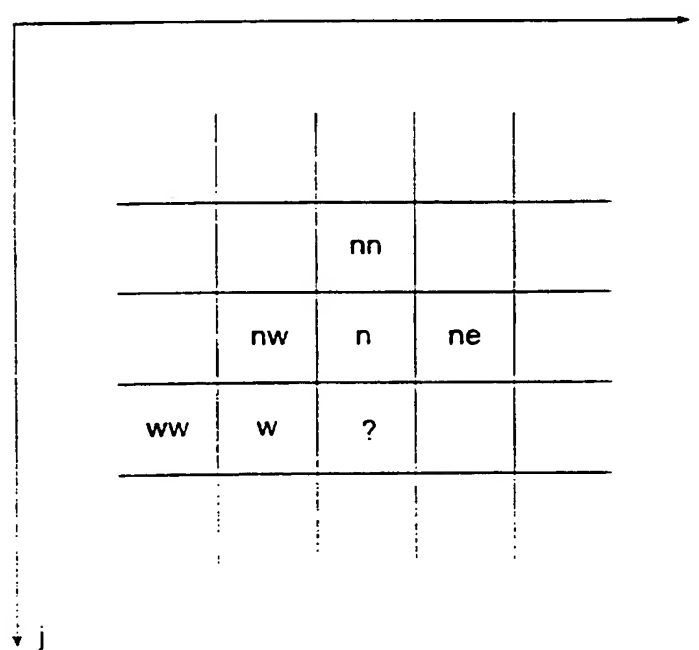


FIG. 2

2/5

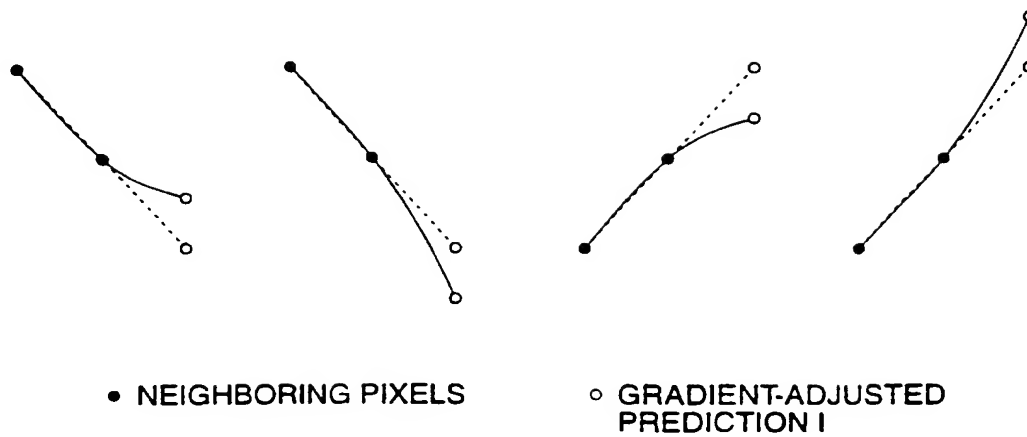


FIG. 3

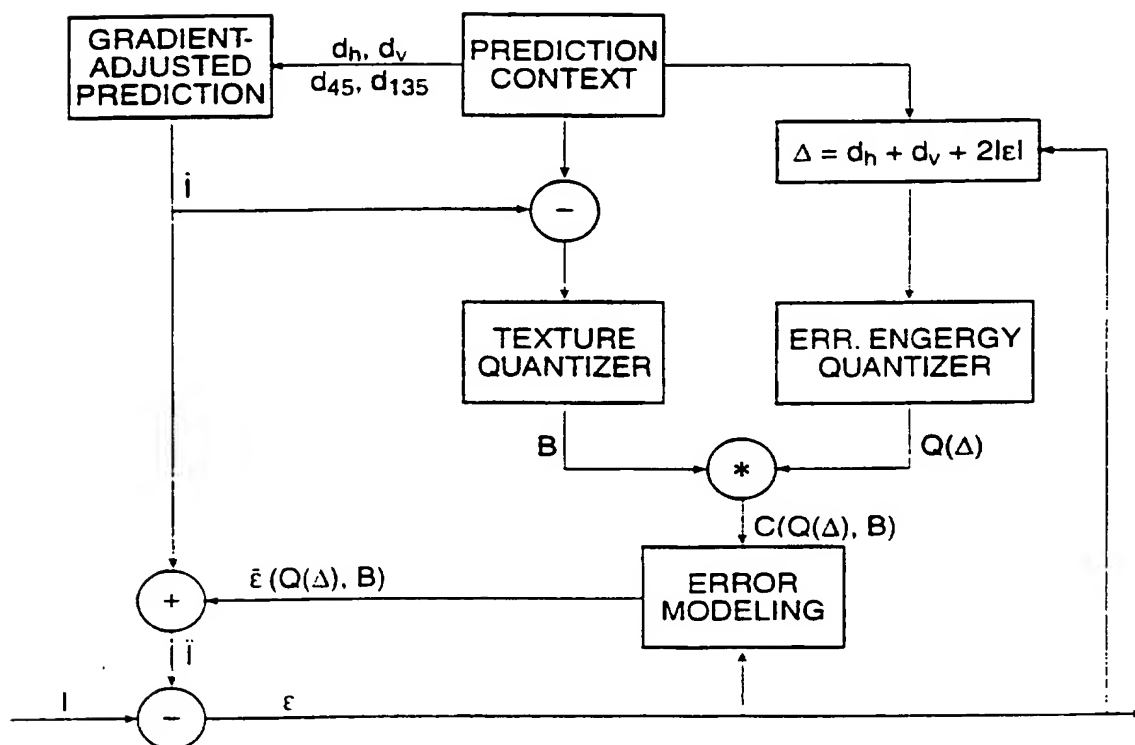


FIG. 4

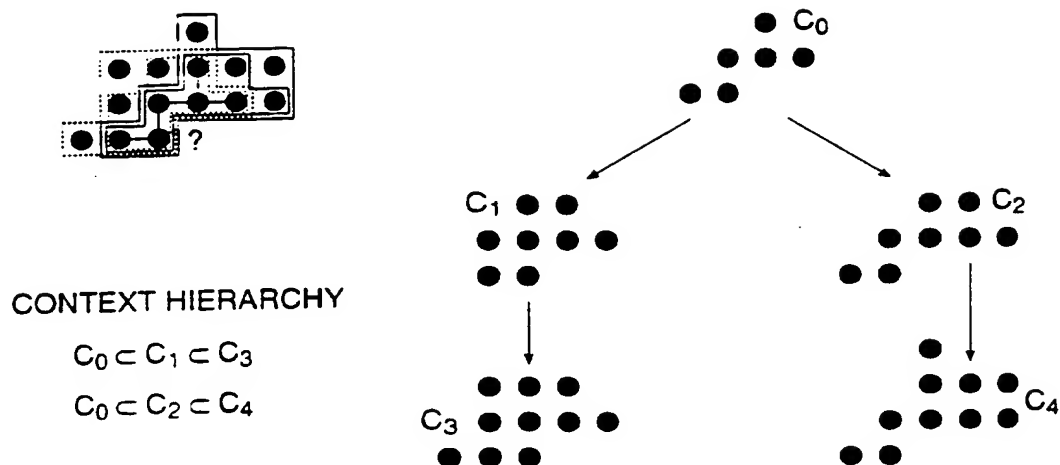


FIG. 5

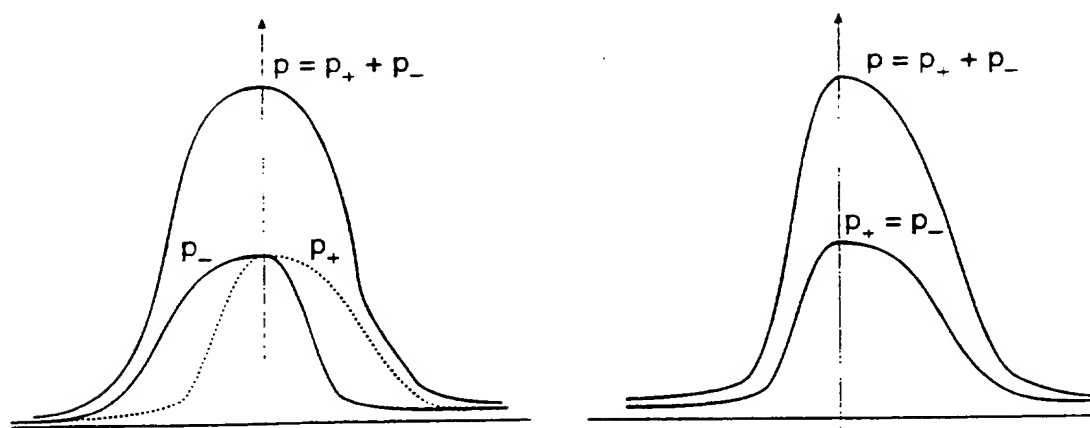


FIG. 6

4/5

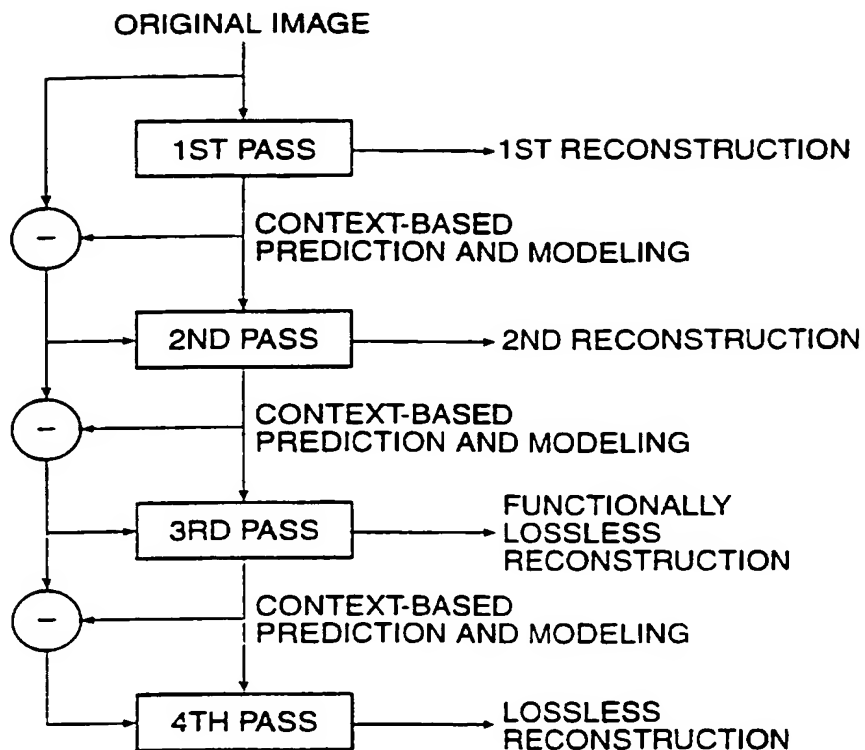


FIG. 7

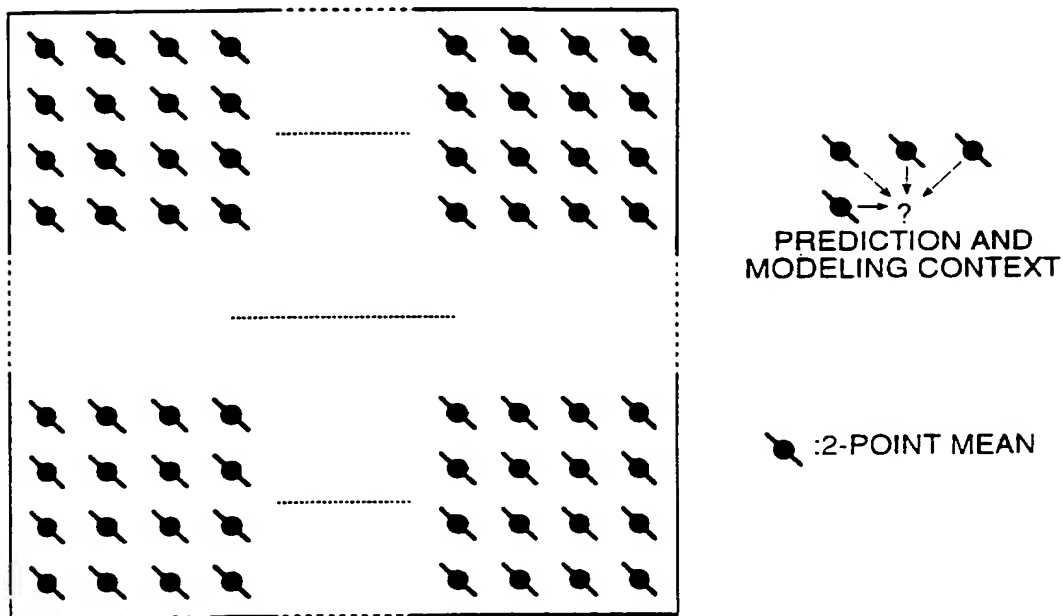


FIG. 8

SUBSTITUTE SHEET (RULE 26)

5/5

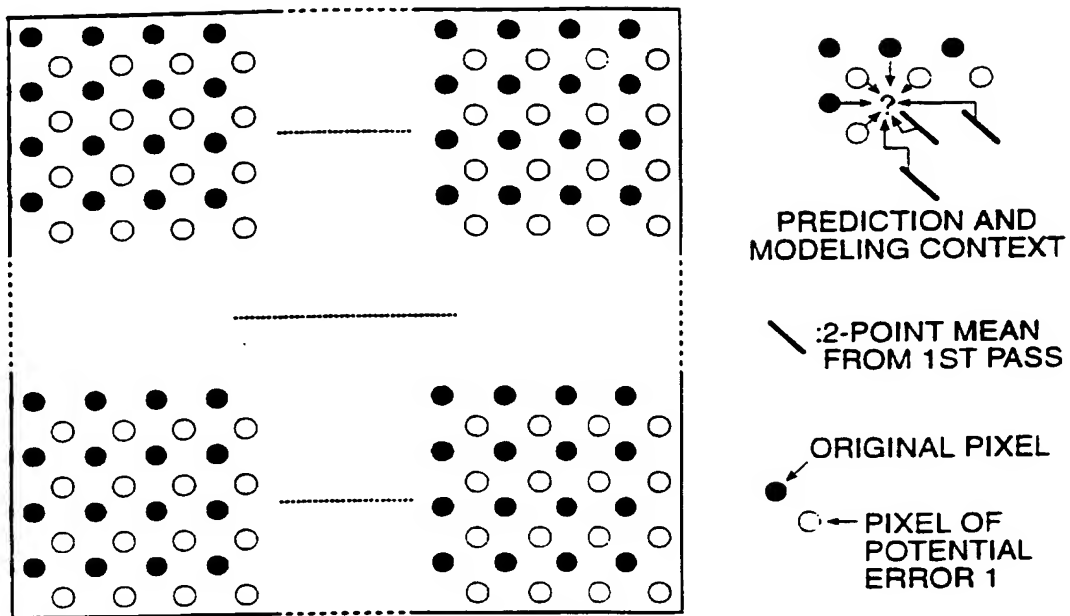


FIG. 9

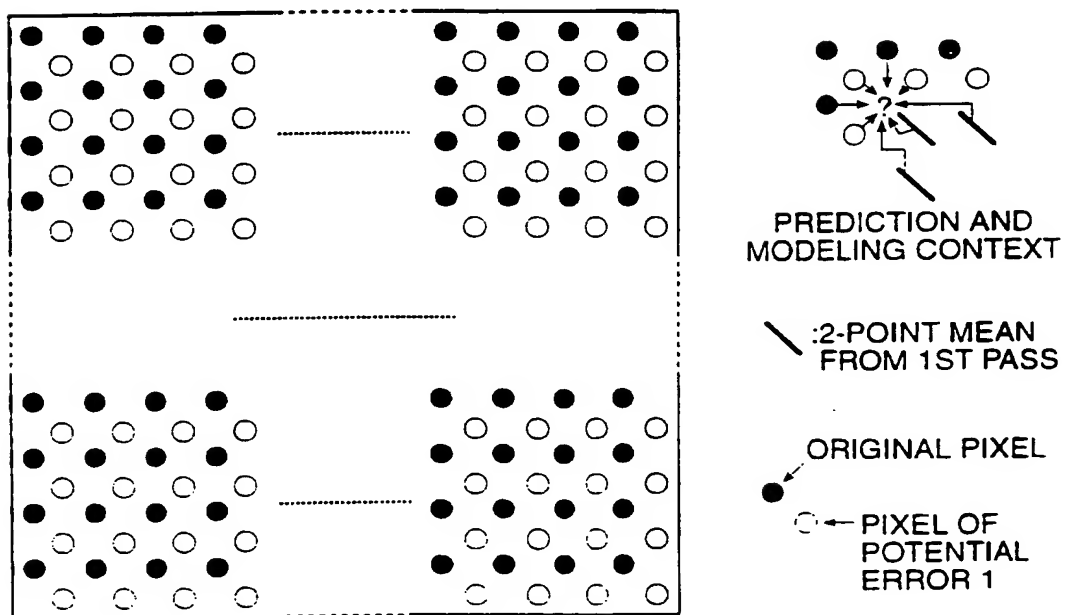


FIG. 10

SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

Intern. Application No.
CA 95/00635A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04N1/41

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 613 290 (NEC CORPORATION) 31 August 1994 see abstract; claim 3; figures ---	1,5,18, 20-23
A	COMPUTER JOURNAL, vol. 36, no. 1, 1993 LONDON GB, pages 68-77, XP 000360266 TISCHER ET AL 'Context-based Lossless Image Compression' ---	1-23
A	EP,A,0 547 528 (EASTMAN KODAK) 23 June 1993 see abstract -----	1-23

☐ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- *G* document member of the same patent family

Date of the actual completion of the international search

6 March 1996

Date of mailing of the international search report

10.04.96

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+ 31-70) 340-3016

Authorized officer

Isa, S

INTERNATIONAL SEARCH REPORT

Intern. al Application No

PCT/CA 95/00635

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-613290	31-08-94	JP-A- 6311372 US-A- 5491564	04-11-94 13-02-96
EP-A-547528	23-06-93	US-A- 5442458 JP-A- 5300382	15-08-95 12-11-93